

# GRACEに於けるソーティングユニットの機能拡張

伏見信也, 喜連川 纈, 田中英彦, 元岡 達  
(東京大学 工学部)

4G-4

## §0. はじめに

ハードウェアソータを用いた関係代数処理の高速化については、従来から数多くの提案、試作が行われているが、実際使用環境に於ては種々のパラメータ(ロード長, ストリーム長等)の変動に対するソータの柔軟性が乏しく、その高速性に対する大きな代償となつてきた。ここではこれらパラメータの変動に対し柔軟に対応できるような GRACE ハードウェアソータの拡張機能について述べる。

## §1. K-way への拡張

試作されたソータ<sup>[1][2]</sup>は 2-way マージソートを行うプロセッサを 1 次元状にパイプライン結合した構成とるが、マージ数を一般に K-way に拡張し、種々の性能向上, 柔軟性を得ることが出来る。この場合も 2-way の場合と同様、メモリ中では各ストリングはロード毎に付加されたポインタを用いて linked list とし管理される。各ストリングの先頭ロードのアドレスを保持するレジスタ (STP<sub>i</sub>: i-th String Top Register) は少なくとも 2K-1 個必要であるが、ここでは制御の容易さからマージ中のストリングに対して K 個 (STP<sub>i</sub>), 次のマージ用の入カストリングに対して K 個 (STP<sub>i</sub><sup>\*</sup>), 計 2K 個を用い、必要に応じてこれらを一括にスイッチして用いることにした。一方、メモリへのアクセス競合を考慮すれば、パイプラインを乱さない為にはメモリから比較器への直接入力は 1 入力に限られる。従つてこの入力を他の入力と分離した構成も可能であるが、ここでは比較器の K 入力全てに MTR<sub>i</sub> (Merge Top Register for i-th string) なるレジスタを設け、全ての MTR<sub>i</sub> の比較器への入力用 latch にメモリからの直接入力が可能な構成とした (図 1)。以上の構成により、2-way の場合と同様、1 byte / 2 clock の完全パイプラインを形成することが出来る。また K-way ではマージ数が動的に変化する為、この為の制御機構を設けてある。

## §2. ロード長変化に対する制御

MTR<sub>i</sub> のレジスタ長を L (ビット), 入カストリムのロード長を X (ビット) とする。X は一般に L に等しいとは限らない。X < L の場合は X = L の場合とほぼ同様の制御を行なうことが出来る。しかし X > L の場合は MTR<sub>i</sub> に入り切らない X-L バイトをメモリに置くことができず、別の工夫が必要である(但し、ロードのキー部は MTR<sub>i</sub> に入り切るものと仮定する)。この場合、メモリへのアクセス競合がおこることはないが、キー部の比較が終了するまで入力されてくるロードを格納するメモリアドレスが定まらない。そこでメモリ中に write buffer 領域を 1 ロード分用意し、書き込み用のメモリアドレスレジスタ MARW を用いて入カロードを常にこの buffer に書き込むこととし、一方、メモリ読み出しには読み出し用のメモリアドレスレジスタ MARR を用いることにした。比較の結果、常に 1 ロードが出力されていく為、write buffer は常に 1 ロード分メモリ中に確保できる (図 1)。

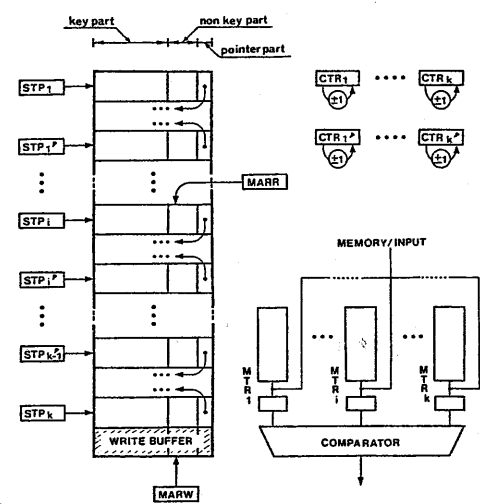


Fig.1 Processor Configuration

## §3. Length Tuning

X ≠ L の場合、各プロセッサのメモリには fraction が生じる。プロセッサ全体のメモリ使用効率は初

段のそれと決つてしまつたが、プロセスの段数  $i$  が進むにつれて fraction も  $K$  倍に増加して  $\dots$  為、 $i \gg 1$  なるプロセス  $P_i$  ではメモリ中に大きな未使用領域が生じてしまう。 $X > L$  の場合はソートのパワ- (ソートすべきロードの最大数) には余カがあるから、この未使用領域を用いることによつてより多くのコードをソートすることにできるよゝに「なる。これを length tuning とする [3]。一般に  $P_{i+1}$  の持つメモリ  $K^{i+1}L$  バイトに長さ  $X$  のコードを  $i+1$  語詰め込み、 $j \geq i+1$  なる  $P_j$  のメモリ使用効率  $E = ([K^{i+1}L/X]X) / K^{i+1}L$  に保つこと  $E$   $i$  次の length tuning とする。一般に  $i \geq 1$  に対して、

$$(*) \quad \left[ \frac{K^{i+1}L}{X} \right] = h_{i1} \left[ \frac{K^i L}{X} \right] + h_{i2} \left\{ \left[ \frac{K^i L}{X} \right] + 1 \right\} \quad (h_{i1} + h_{i2} = K, h_{i1}, h_{i2} \geq 0)$$

なる  $h_{i1}, h_{i2}$  が定まる。即ち  $i$  次の tuning を行つた為には前段のプロセスは長さ  $[K^i L/X]$ ,  $[K^i L/X] + 1$  のストリングを各々  $h_{i1}, h_{i2}$  回送出すればよい。従つて  $1 \leq j \leq i$  なる  $P_j$  のメモリには高々  $[K^j L/X] + 1$  のロードを収めればよいことになる。 $L < X \leq K \cdot L$  に対して  $P_j$  に必要なメモリ容量の最大値は次式で与えられる。

$$\frac{(K^{j-1} + 1) K^{j+1} L}{K^j + 1}$$

従つて  $P_j$  に余分に付加すべきメモリと元のメモリとの比は  $(K-1)/(K^j+1)$  とする。 $j \gg 1$  の時はほとんど無視できることになる。一方、(\*) とその右辺に対して再帰的に適用することによつて、 $i$  次の tuning を行つた際には初段のプロセス  $P_1$  のみで  $[K^i L/X]$ ,  $[K^i L/X] + 1$  way の 2 種類のマ-ジを行つて十分であることがわかる。以上の length tuning に対する制御を行つた

は  $P_1$  に対してマ-ジの way 数  $k$  を、 $P_j$  ( $2 \leq j \leq i$ ) に対して入カストリングの長さ  $X$  を指示する必要がある。ここでは前者に対しては eol, eol flag (end of load) を  $k-1$  番目のストリングの最後のロードに、後者に対して

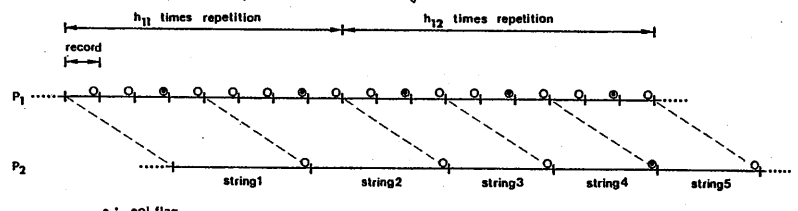


Fig.2 eol & eost Flag Manipulation For Length Tuning

では各ストリング毎に最後のロードに eost flag (end of string) と各々付加することにした。図2に  $K=5, h_{11}=2, h_{12}=3$  の場合のタイミングチャートを示す。

§4. ストリーム長の変動に対する制御

ソートのパワ-を  $M$  (ロード), 入カストリングの長さを  $Y$  (ロード) とする。 $Y > M$  の場合はソートを複数回制御する工夫が必要となる。一方、 $Y < M$  の場合、 $i > \lceil \log_K Y \rceil$  なる  $P_i$  について最早入カロードを素通しさせるだけである。この制御の為に eos (end of stream) 及び eol (end of record) の 2 つの flag を設計した。eol はストリームの最後のロードに付加される。一方、eos は入カストリング中の最後のストリングの先頭ロードに付加され、マ-ジを行つた度にこの条件を満足する要素、bubble up される。この時、上記の  $P_i$  に於て付加されるストリームの先頭ロードに eos が付加されていることになり、以後 eol を検出するまで素通しを行つてよい。

§5. おわりに

現在、ストリーム内のロード長が変化する場合の最も一般的で使用環境に対して Block Division Method [3] と呼ぶた解決を考慮中である。

[参考文献]

[1] 伏見 他 「可変構造多重処理データベースマシンに於けるソートユニット」 情報処理 23 回全国大会。  
 [2] 桑原 他 「GRACE に於けるソートユニットの構成」 情報処理 24 回全国大会。  
 [3] 喜連川 他 「可変構造多重処理データベースマシンにおけるソートモジュール」 EC&I-15