

パイプラインマージソータに於ける可変長レコードソート用

5H-3

String Length Tuning アルゴリズムとその評価

楊 維康 喜連川 優 高木 幹雄

東京大学 生産技術研究所

1. はじめに

VLSI技術によって汎用性のあるアルゴリズムを専用ハードウェアで実現する場合、そのハードウェア化による高速性を追求すると同時に、実用時のパラメータの変動に対応すべく、柔軟性を持たせることも重要である。我々はこうした目的を目指してパイプラインマージソータの([1][2])研究開発を続けてきた。

ハードウェアソータに於て重要なパラメータの一つはレコード長である。そのレコード長の変化には、ファイル内では固定長レコード(ファイル内では固定でファイル間では可変)、及び可変長レコード(ファイル内でも可変)、という二つの状況が考えられる。前者に対しては、我々はすでにString Length Tuningアルゴリズムを開発し([6])、ハードウェアで実装した。更に、後者についてはソータの記憶管理方式について詳しく考察し、ブロック分割記憶管理方式によって、可変長レコードソートをサポート出来ることを示した([3][4])。本稿では、可変長レコードソータに於て、ソータのメモリを効率的に管理できるString Length Tuningアルゴリズムを紹介し、その性能を評価する。

2. 可変長レコードソータの実現の問題点

パイプラインマージソータ(2-Way)の基本アルゴリズムは、一次元に結合される各ソートプロセッサが初段から入力されてくるストリングを2本ずつマージソートしていくのである。これにより*i*番目のプロセッサ P_i が出力するストリングは 2^i レコードからなる。

可変長レコードソータの場合、レコード長 x は確率変数であり、 P_i の出力するストリング長 L_{ij} ($j=1, 2, \dots, \lfloor \frac{M}{2^i} \rfloor$)も確率変数となる。アルゴリズム上、 P_{i+1} は 2^i レコードのストリングをメモリにロードする必要があるので、プロセッサ P_{i+1} のメモリ M_{i+1} は、 $M_{i+1} > \max_j(L_{ij})$ の記憶容量が必要となる。しかし、実際のデータに対して $\max_j(L_{ij})$ の計算は困難であり、又、ストリングの最大値に合せてメモリを用意することは明らかに非効率的である。可変長レコードのストリームを効率良くソートする為には、プロセッサが一定のメモリ容量を有し、不確

定な長さのデータに対処できる記憶管理アルゴリズムの確立が必要である。

3. 可変長レコードソータに於ける String Length Tuning

可変長レコードソータに於ける問題はレコード長の変動により、各プロセッサの生成するストリングの長さがデータに依存し、一定でないことにある。この問題を解決する為に、我々の基本的な発想は、一次元に結合された*n*個のソートプロセッサのうち、先頭から*d*個の P_1, P_2, \dots, P_d に於てマージソート処理時に適当な制御を加え、可変長レコードによるストリング長の不確定さがある程度吸収し、比較的長さが安定したストリングを作り出す。これにより、 P_d 以降のソートプロセッサは長さがほぼ安定したストリングを取り扱うようになる為、メモリ利用効率低下の問題を大幅に改善できる。本論文ではこのことを可変長レコードソータに於けるString Length Tuning(SLT)と言う。SLTを行う為には P_1, P_2, \dots, P_d 於て補助メモリが必要であるが、パイプラインマージソータに於て、最初の数段のプロセッサはメモリ容量が少なく、ソータ全体のメモリ利用効率は、ほぼ P_d 以降のプロセッサのメモリ利用効率によって決定される。

具体的な手法はデータストリームを入力と同時に先頭から複数のサブストリームに分割する。ストリームの分割は、条件

$$\begin{aligned} L_{\text{SUB}} &\leq 2^d \cdot L \\ n_{\text{SUB}} &\leq 2^d \end{aligned}$$

下で、 L_{SUB} が最大であるように行われる。ここで、 L_{SUB} はサブストリームの長さで、 n_{SUB} はサブストリームのレコード数である。

つぎに、 P_1, P_2, \dots, P_d では、それぞれ $M_i = 2^{i-1} M_1$ のメモリを持ち、各サブストリームをソートする。 M_1 は P_1 のメモリ容量である。 P_i は以下のように動作する。

P_i はレコードの入力にあたって、そのレコードが当該メモリに収容できるか否かを判断する。収容できる場合はマージ処理をし、収容できなければ、それをバイパス

String Length Tuning Algorithm for Variable Length Record on Pipeline Merge Sorter

Weikang YANG, Masaru KITSUREGAWA, Mikio TAKAGI

Institute of Industrial Science, University of Tokyo

する。一つのサブストリームの入力が終了すると、メモリ内残存の当サブストリームのデータを出力する。

P_d の出力するストリングの長さは $2^d L$ 以下であり、 P_d 以降のプロセッサ P_i ($i > d$) は $2^{i-d} L$ のメモリによりマージソートを行う。

文献 [5] において、本アルゴリズムの正当性を証明し、 $M_1 = 2L$ であることを示した。つまり、本アルゴリズムにより、 P_1, \dots, P_d が固定長 (L) のレコードソート時の2倍のメモリ空間を持つことにより、可変長レコードソートの S L T を行うことができる。

4. S L T によるソートのメモリ利用効率

可変長レコードソートの場合、レコード長 x が確率変数であり、サブストリーム長 L_{SUB} も確率変数である。それらの間の関係は複雑であり、一般的な定量解析は困難である。その為、まずシミュレーションで、S L T アルゴリズムによるメモリ利用効率を評価した。

一般的にメモリ利用効率 η は以下のパラメータによって決定される。

n : プロセッサの台数

L : ソートの設計レコード長

d : Length Tuning の次数

$p(x)$: レコード長の確率密度関数で、レコード長の平均及び分散を決定する。

レコード長 x が平均 μ , 分散 σ である正規分布

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

に従うとする。 $\eta = f(n, d, \mu/L, \sigma)$ である。

まず $\mu/L = 1, \sigma = 100$ として、プロセッサ台数 $n = 12, 14, 16$ 時の S L T の次数 d と η の関係を図1に示した。図の横軸が S L T の次数 d であり、縦軸がメモリ利用効率 η である。図に示したように、プロセッサ16段結合のソータは、次数8、或いは次数9の S L T を行う時、メモリ利用効率はほぼ最大値に達する。

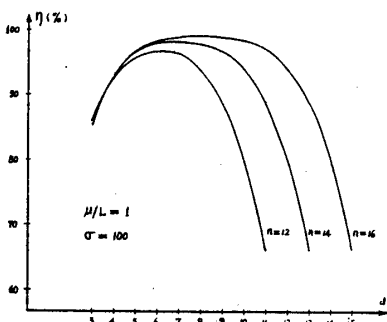


図1. d と η の関係 (パラメータ n)

次に、 $\mu/L = 1, n = 16$ である時、 σ, d と η の関係を図2に示す。図から分かるように S L T の次数 d が比較的小さい時、 η は σ より変動するが、 d が比較的に大きい場

合、 σ による変動は殆どない。これは d が大きくなると tuning の空間が大きくなり、レコード長のばらつきを吸収することによる。

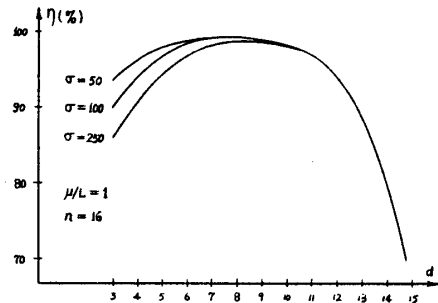


図2. d と η の関係 (パラメータ σ)

図3には、 $n = 16, d = 8$ である時、 μ/L と η の関係を示す。図からわかるように $\mu = L \sim 16L$ の間、 η がほぼ98%を保つ。

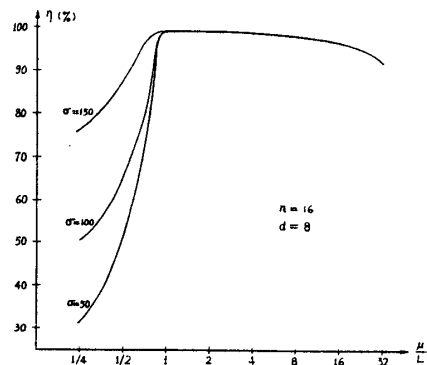


図3. μ/L と η の関係 (パラメータ σ)

7. おすび

以上、パイプラインマージソータで、可変長レコードを効率良くソートするString Length Tuningアルゴリズムについて紹介した。我々は本アルゴリズムの実装法について検討し、レジスタトランスフェレレベルのアルゴリズムを記述するとともに、C言語で書かれたシミュレータでアルゴリズムの正当性を確認している。

<参考文献>

- [1] 喜連川 電気通信学会論文誌, J66-D 1983年 3月
- [2] 伏見他 情報処理学会第24回全国大会 4G-5 1982
- [3] 楊他 情報処理学会第31回全国大会 1B-9 1985
- [4] 楊他 情報処理学会第32回全国大会 3C-5 1986
- [5] 楊他 電子通信学会技術研究報告 SE86-15
- [6] M.Kitsuregawa, Memory Management Algorithms of Pipeline Merge Sorter, International Workshop of Database Machines, Florida (1985)