超並列計算機におけるデータ並べ替えアルゴリズムと要求されるデータ転送能力の見積もり

佐藤隆士† 津田孝夫††

多階層共有メモリをもつマルチ CPU マシンをモデル化し，データ転送ネックにならないために必要な転送能力を求めている．具体的適用例として，データ並べ替え問題に的を絞り，処理時間が単一 CPU マシンの場合の下界に対して，CPU 台数分の1で並べ替えする並列アルゴリズムを提案している．また，細粒度並列処理への適用を考慮し，並列処理可能な CPU 台数の上限を与え，データ数が n で，n に比例する台数の CPU を用いる場合，計算時間およびデータ転送量が高々 $\log n$ の定数倍であることを示している．さらに，CPU の計算能力を最大限に発揮させるために必要なレベル間のデータ転送能力，各レベルの記憶容量を見積もっている．

Estimation of Data Transfer Rate Required for Permuting Data on a Massively Parallel Processor

TAKASHI SATO† and TAKAO TSUDA††

A massively parallel processor having multi-level shared memory is modeled to find a memory bandwidth between PEs (processor elements) such that it does not give rise to a data-transfer bottle-neck. Focusing on concrete data-permuting problems, this paper also gives a parallel algorithm whose processing time is the lower bound of a single-processor time cost times the inverse of the number of PEs. Concerning fine grain parallel processing, the upper limit number of PEs which can process data in parallel is given. When the number of data is n and PEs of which number is proportional to n are used, this paper shows that its computation time and the amount of data transfer could be constant times of $\log n$. Interlevel data-transfer rate of the hierarchical memories and the capacity of memory at each level are also derived to maximize the PEs computing power.

1. まえがき

今後，並列処理計算システムは，FORTRAN-90, HPF (High Performance Fortran)[1]等の並列処理指示可能な言語で書かれたプログラムを特に高性能に処理できなければならない．これに関し，3次元数値計算を行う際，次のような問題を能率よく処理する必要性が報告されている[2]．

（1） 配列データを第1次元についてマルチ CPU マシンの各 CPU のローカルメモリに配り，数値計算を行う．

（2） 次に，もとの第2次元を第1次元になるように転置を行い，データを各 CPU のローカルメモリに配りなおし，数値計算を行う．

（3） 次に第3次元について同様に行う．

（1），（2），（3）を繰り返しながら数値計算を進めていく．この際，マルチ CPU 環境では，数値計算そのものだけでなくローカルメモリへのデータの配りなおしにかかる時間が問題となる．HPF では，プロセッサへのデータ割付けをプログラマが明示的に指定できるので，配列データの割付け変更を行うと，データの配りなおしが必要となる．このようなデータの配りなおしは，計算量は少ないが共有メモリを介してローカルメモリ間で行うデータ転送量が多い典型例で，現存する多くのマルチ CPU 計算機では，データ転送ネックになることが予想される．

データの配りなおしはデータの並べ替え問題に帰着される．記憶階層下におけるデータ並べ替えの先駆的な研究は，Floyd[3]による．一方，著者らは，単一 CPU マシンにおけるデータの並べ替えについて，2レベルあるいは3レベル以上の記憶階層環境において，論理的なデータ転送量の下界を実現するアルゴリ

† 大阪教育大学教養学科
Department of Arts and Sciences, Osaka Kyoiku University
†† 京都大学工学部
Faculty of Engineering, Kyoto University

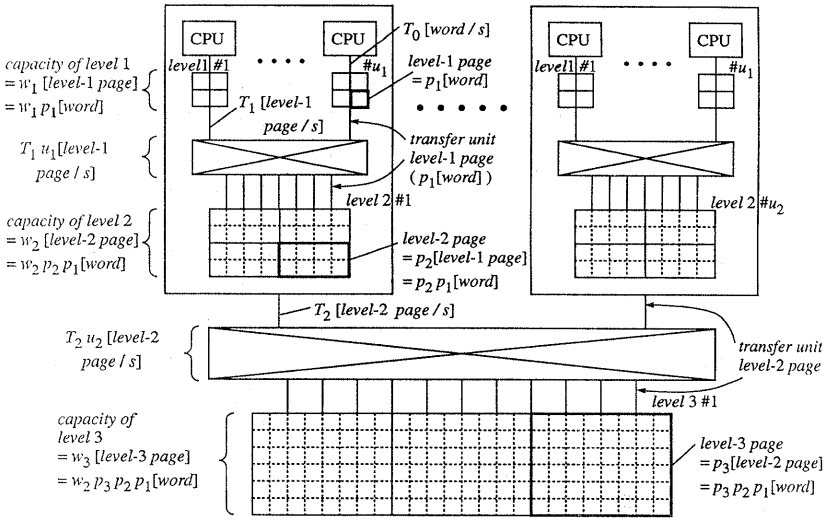


図 1 多階層記憶をもつマルチ CPU マシンモデル ($m=3$)
Fig. 1 A model of multi CPU machine ($m=3$).

ズムを提案している[4),5)]. また, Aggarwal と Vitter[6)] は, 並列にアクセス可能な二次記憶をもつ単一 CPU モデルで, 並べ替えに関連したいくつかの問題に対して, 最適アルゴリズムを与えている. 本論文では, 以前著者らが発表したアルゴリズムを, 多階層記憶をもつマルチ CPU マシン[7)]を対象に, 拡張したアルゴリズムを提案する.

このアルゴリズムにより, 単一 CPU マシンの場合の下界処理コストをマルチ CPU マシンの各 CPU に均等に配分でき, 処理時間を CPU 台数分の1にすることができた. また, 細粒度並列処理への適用を考慮し, モデルパラメータに対して並列処理可能な CPU 台数の上限を与えた. その結果, データ数が n のとき, 並列処理可能な CPU 台数を n に比例させることができ, CPU あたりの計算時間およびデータ転送量が高々 $\log n$ の定数倍となることがわかった.

現在, 製作されている商用並列計算機は, ローカルメモリおよび共有メモリの2レベル, またはローカルメモリを直接バスあるいはスイッチングネットワークで接続した1レベルの記憶階層をもつものなどであり, 記憶階層のレベルは多いとは言えない. しかし, プロトタイプ段階では3レベル構成のものも試作されている[8)]. 今後, 集積回路の集積度がさらに高まり, 1個の IC 内に複数台の CPU を搭載するようになると, IC 内部において, CPU のローカルメモリだけでなく共有メモリをもつと予想される. IC 外部の記憶レベルと合わせて, 今後3レベル以上の記憶階層をもつ超並列計算機が出現するものと思われる.

以下, 2章では多階層のローカル・共有メモリをもつマルチ CPU マシンをモデル化する. 3章では, 一般的な計算について, 各レベルのメモリに読み込まれてからのデータの平均参照回数をパラメータに, マルチ CPU の最大計算能力を実現するために要求される, メモリ間のデータ転送能力を見積もる. さらに, 4章では具体的なデータ並べ替え問題に的を絞り, 考察を行う. すなわち, マルチ CPU マシンに対する具体的アルゴリズムを与え, データ転送コスト, 要求されるデータ転送能力, 必要なメモリの大きさについて解析する. 4章では3階層記憶構成について述べるが, 5章では m 階層記憶構成の場合に拡張し, その結果を示す. 6章はむすびである.

2. 超並列計算機モデル

本論文で扱う超並列計算機モデルは, クロスバなどのスイッチングネットワークで結合された, 多階層記憶をもつマルチ CPU 構成である. **図1**に階層数 $m=3$ の場合を示す. レベル1は, キャッシュ (またはローカルメモリ) をもつ CPU である. CPU-キャッシュ間のデータ転送単位は語 (0次ページ) で, CPU の処理能力 (または CPU-キャッシュ間の転送能力で計算が制限されるときはその能力) を, T_0[語/s] とする. ただし, [ ] の内容は単位を表す. 各キャッシュの記憶容量は,

$$W_1 = w_1[\text{1次ページ}] = w_1 p_1[\text{語}] \qquad (1)$$

とする．ここで，p_1 は，レベル 1-2 間の転送単位を語で表したもので，1[1次ページ]$=p_1$[語] である．レベル2は，記憶容量

$$W_2=w_2[\text{2ページ}]=w_2p_2[\text{1次ページ}]=w_2p_2p_1[\text{語}] \quad (2)$$

をもつ記憶装置であり，レベル1の u_1 台の CPU およびキャッシュとスイッチングネットワークで接続されている．この間のデータ転送能力は，レベル1の装置1台当り，T_1[1次ページ/s] であり，スイッチングネットワークにより同時に u_1 台のキャッシュにあるいはからデータ転送可能であるとする．このため全体では，

$$T_1u_1[\text{1次ページ/s}] \quad (3)$$

の転送能力をもつ．この間の接続は，T_1u_1 [1次ページ/s] の転送能力をもつバス結合の場合であっても，バスを時分割使用することにより，スイッチングネットワークと同等の機能を果たすことができる．

レベル 2-3 間，レベル3以上についても同様に定義できる．一般に，レベル i の u_i 個の装置をレベル $(i+1)$ にスイッチングネットワークで接続する．この間の転送単位は i 次ページ，転送能力は1経路当り，T_i[i次ページ/s] であり，レベル i の記憶容量は w_i[i次ページ] である．また，i 次ページの1ページに $(i-1)$ 次ページがちょうど p_i 個格納可能であるとする．

なお，最下位にあるレベル m の記憶装置を省略し，直接レベル $(m-1)$ の装置間をスイッチングネットワークで接続することもできる．この構成をとる場合は，レベル m にデータを保管するための場所がなくなるため，レベル $(m-1)$ に要求される記憶容量はより大きくなる．レベル $(m-1)$ 間のスイッチングネットワークの転送レートは，1経路当り T_{m-1} のままでよい．最近発表された並列計算機，富士通 VPP 500[9] は $m=2$，DASH[8] は $m=3$ のこのタイプの記憶構成をもつ計算機と見ることができる．

次章以降では，可能な限りすべての CPU を並列に動作させることを前提に，要求されるデータ転送能力を見積もる．

3. 要求されるデータ転送能力

各メモリ階層間で要求される転送能力は，計算プログラムのメモリ参照パターンにより異なる．本論文では，レベル $(i+1)$ $(i=0,1,\cdots,m-1)$ にあるページが1つ上のレベル i に読み込まれて，そのページがレベル i に保持される間に，参照される平均回数をパラメータ $R_i(\geq 0)$ とする．レベル i-$(i+1)$ 間がデータ転送のボトルネックにならないためには，レベル $(i+1)$ の1装置当り u_i 台のレベル i の装置が接続されているので，

$$T_{i+1}=T_iu_i/(R_ip_{i+1})[(i+1)\text{ 次ページ/s}] \quad (4)$$

以上でなければならない．ここで，$u_0=1$ であり，$1/p_{i+1}$ はページ転送単位の変換である．すなわちレベル i-$(i+1)$ 間で要求される転送能力は，レベル i から上を見た全転送能力の $1/R_i$ である．上位レベルで使用されないページは転送しないとしても，記憶参照の局所性がない場合は $R_i=1$ となるので，大容量を要求される下位 $(i+1)$ レベルの記憶にとってはきびしい条件になる．

レベル i-$(i+1)$ 間の転送特性を単純に1次関数で，

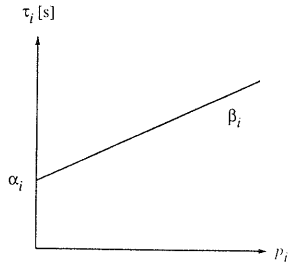
$$\tau_i=\alpha_i+\beta_ip_i[\text{s}] \quad (5)$$

で表すことにする（**図2**（a）参照）．ここで，p_i は $(i-1)$ 次ページで測った転送単位である．α_i[s]，β_i[s/$(i-1)$ 次ページ] は定数，τ_i は1つの i 次ページを転送するのに要する時間である．このとき転送スピードは，

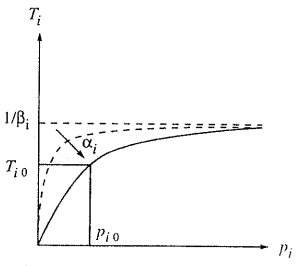
$$T_i=p_i/\tau_i=p_i/(\alpha_i+\beta_ip_i)\ [(i-1)\text{ 次ページ/s}] \quad (6)$$

である．これを p_i について解くと，

$$p_i=\alpha_iT_i/(1-\beta_iT_i) \quad (7)$$



（a） 転送単位-転送時間



（b） 転送単位-転送スピード

図2 データ転送特性
Fig. 2 Characteristics of data transfer.

となる．$p_i>0$ より，$1/\beta_i>T_i$ である必要がある．図2(b)に示すように転送スピード T_{i0} を必要とするときは，p_{i0} 以上のページサイズで転送する必要がある．α_i は小さくなると，同図破線のカーブのように変化する．特に，$\alpha_i=0$ のときは，p_i は最小単位の1，すなわち i 次ページは $(i-1)$ 次ページと同じでよい．

4. 3階層モデルにおけるデータ並べ替え

ある特定のアルゴリズムへの適用例として，データ並べ替えについて考察する．いわゆるソーティングと異なり，初期状態から最終状態（目標状態）への要素と移動先があらかじめわかっている場合である．1章で述べた，多次元配列に格納されたデータを異なる次元方向に配りなおす問題は典型例である．並べ替えは，CPU内での計算負荷が軽い問題であるため，CPU の能力を生かすためには，高いデータ転送能力が要求される．データ転送コストとしては，レベル i-$(i+1)$ 間のデータ転送量を転送単位である i 次ページの読込みページ数で見積もる．初期並びと最終目的とする並び間の隔たりによりデータ転送量は異なるが，本論文では配列データの転置のように最も転送コストが大きくなる場合を上限とした式で表現している．また，配列データ以外の非均質なデータ並べ替えを扱うこともできるが，多次元配列データの異なる次元方向への配りなおしなど，均質データ並べ替えの応用範囲は十分広いこと，および議論を単純でより明解なものにするため本論文では均質で固定長の場合のみを扱うことにする．まず，3階層の場合について詳述し，5章で一般の m 階層の場合に拡張する．

並べ替えるべきデータは，レベル3に格納されている p_3'[2次ページ] であるとする．単一 CPU マシンの場合は，レベル 2-3 間の2次フェッチ数（レベル3から2への2次ページの読込み回数）が，

$$F_2\leq p_3'\lceil\log_{w_2} p_3'\rceil \qquad (8)$$

レベル 1-2 間の1次フェッチ数（レベル2から1への1次ページの読み込み回数）が，

$$F_1\leq p_2\lceil\log_{w_1} w_2\rceil F_2 \qquad (9)$$

で並べ替え可能であることがわかっている[5]．上2式で等号となるのは，初期状態から最終状態への並べ替えが最もランダムな場合である．ここで，最もランダムとは，初期状態で，レベル3の同一2次ページにある各要素が最終状態では，すべて異なる2次ページに移動させられる場合である．1つの2次ページは p_1p_2[語] なので，1要素を1語にとると，2次ページの名前の付け替えを適当に行うことにより，正方行列 p_1p_2 行 p_1p_2 列の2次元配列データを転置することと同じになる．ここで，1行は1つの2次ページに格納されており，行数は総ページ数となるため，$p_3'=p_1p_2$ である．文献5) のアルゴリズムは，単一 CPU マシンでコスト最小の転置を行うことができることがわかっている．

本章では，マルチCPUマシンにおいても，式(8)，(9)と同じページフェッチ数で要素を並べ替えるアルゴリズムを示すと共に，必要とされるデータ転送能力，各レベルの記憶容量を見積もる．

4.1 アルゴリズム

(A) レベル 2-3 間のデータ転送

レベル3にある p_3'[2次ページ] に格納されているデータをレベル2の u_2 台の w_2[2次ページ] で並べ替える．まず p_3' が w_2 のべき乗の場合について説明し，後でそうでない場合に拡張できることを示す．具体的な方法は，単一 CPU マシンでの並べ替えアルゴリズムである $d\log_w d$ 法[4),5)]を拡張したものである．

並べ替えは，$t_2=\log_{w_2} p_3'$ の段階からなる．これを（レベル 2-3 間の）第1から第 t_2 パスと言う．レベル2の u_2 台の装置に負荷を分散させるため，各パスにおいて p_3'[2次ページ] をなるべく均等に p_3'/u_2 ずつ分担させ，並列処理させる．

[定義] 第 j パス $(j=0,1,2,\cdots,t_2)$ のグループ $g(g=0,1,\cdots,w_2^j-1)$ とは，最終状態の要素の並びで，$\{g, w_2^j+g, 2w_2^j+g,\cdots, p_3'-w_2^j+g\}$ の番号の2次ページに含まれる要素のみからなる2次ページのことである．同じグループに含まれるページ数をグループ長と言う． □

第 i パスでは，第 $(i-1)$ パスのグループ g を第 i パスのグループ $g, w_2^{i-1}+g,\cdots,(w_2-1)w_2^{i-1}+g$ に分割する．ここで，$i=1,2,\cdots,t_2$，第0パスのグループは1つだけでその要素は並べかえ対象となる全2次ページ $(0,1,\cdots,p_3'-1)$ である．第 $(i-1)$ パスのグループ数は w_2^i，グループ長はすべて p_3'/w_2^{i-1} である．1パスごとにグループ数は w_2 倍，グループ長は $1/w_2$ になる．この場合，4.2節で述べるように，p_3'/u_2 が w_2 の整数倍であれば，u_2 台の装置へ無駄なく処理を分散することができる．

次に，分割の方法を述べる．第 $(i-1)$ パスのグループ g について，各装置の分担ページをレベル3からレベル2に順次フェッチする．フェッチされた各2次

ページはレベル1を用い，(B)で述べる方法で分解再編成され，第 i パスのグループに組み立てられる．各装置のレベル2の容量だけ2次ページが読み込まれると，w_2 個の2次ページが収まるが，これにより少なくとも1つの第 i パスの2次ページができる（文献4)の補題参照）ので，それをレベル3に追い出す．それがグループ $jw_2^{i-1}+g$ ($j=0,1,\cdots,w_2-1$) に属するとき，次のパスではこのグループの番号で参照できるようページの呼びかえを行っておく．この操作を第1から第 t_2 パスまでのすべてのグループ g について行うと，レベル3の1次ページの並びは最終状態になる．

レベル間のデータ転送量は，読込み（レベル3から2への転送）と書込み（レベル2から3への転送）が同じなので，以下では読込み量のみで評価する．1パス当りのデータ転送量は p_3' なので，全データ転送量は，$F_2=p_3't_2$[2次ページ] となる．

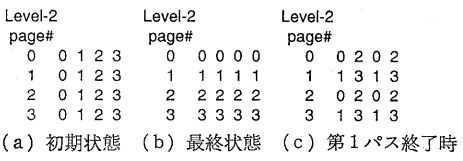
p_3' が w_2 のべき乗でないときでも，2次ページの番号が $p_3'-1$ 以下であることに注意すれば，同じアルゴリズムが使用できる．このとき，パス数は，$t_2\equiv[\log_{w_2} p_3']$ になる．

(B) レベル 1-2 間のデータ転送

レベル 2-3 間の第 i パスではレベル2の w_2[2次ページ] に第 $(i-1)$ パスのグループ $g(g=0,1,\cdots,w_2^{i-1}-1)$ が読み込まれているので，これを(A)で述べたように w_2 個のグループに分割しなければならない．この作業は，各要素（0次ページ）がグループ別に w_2 種類あるので，それを種類ごとに集めることであるとも言える．そのとき必要となる1次ページ内の要素並べ替えは，CPU とレベル1間で行われる．この並べ替えにも $d\log_w d$ 法が有効である．この場合，レベル 1-2 間の並べ替えのパス数は，$t_1\equiv\lceil\log_{w_1} w_2\rceil$ となり，(A)と同様に定義されたレベル 1-2 間のグループについて，パスごとにグループ分割を行う．$w_1>w_2$ のとき，w_1-w_2 は使われない．第 i パスでは，第 $(i-1)$ パスのグループ g を第 i パスのグループ $g, w_1^{i-1}+g, \cdots, (w_1-1)w_1^{i-1}+g$ に分割する．この場合についても，4.2 節で述べるように，w_2p_2/u_1 が w_1 の整数倍であれば，u_1 台の装置へ無駄なく処理を分散することができる．

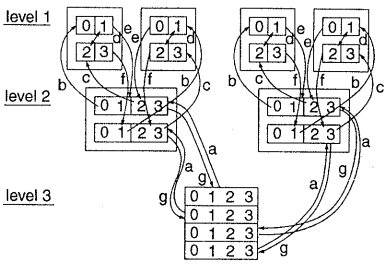
この並べ替えは，レベル2に読込まれた全ページ数 F_2 について行われる．レベル 1-2 間の転送は1次ページで行われるので，全データ転送量は $F_1=p_2t_1F_2$ となる．

[例1] 各要素を最終状態で移動させられるべきレベル3の2次ページ番号で表し，レベル3の初期状態を**図3**(a)，最終状態を同図(b)とする．パラメータは，$p_3'=4$, $p_1=p_2=2$, $w_1=w_2=2$, $u_1=u_2=2$ とする．4×4 の2次元配列データを CPU 4台，レベル2の装置2台で行う転置例である．レベル2のパス数は，$t_2=2$ である．第1パス終了時には，図3(c)に示すようになる．第2パス終了時には，同図(b)のようにならなければならない．第1パスは，$g=0$ のグループ ({0, 2}) と $g=1$ のグループ ({1, 3}) に分割する．ここで，{ } 内は，最終状態で移動させられるべきレベル3の2次ページ番号で表した要素の集合である．並べ替えの詳細を**図4**(a)に示す．第2パスは，{0, 2} からなるグループを{0}と{2}に，{1, 3} からなるグループを{1}と{3}にそれぞれ分割する．その詳細は同図(b)に示したとおりである．四角は各レベルのページを表しており，縦線で区切られた領域は上位レベルとの転送単位である．図中，矢印はデー

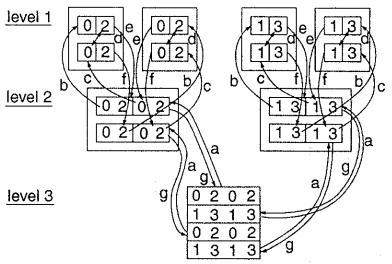


(a) 初期状態 (b) 最終状態 (c) 第1パス終了時

図 3 3階層マルチ CPU マシンでの並べ替えの例
Fig. 3 Example of rearranging elements on 3-level multi CPU machine.



(a) 第1パス



(b) 第2パス

図 4 並べ替えの詳細
Fig. 4 Details of rearranging elements.

タの移動（コピー）を示す．レベル1内では要素単位，レベル 1-2 間では1次ページ単位，レベル 2-3 間では2次ページ単位である．移動の順番は，矢印に付けられた a, b, c, d, e, f, g の順である．なお，同じ記号で示した順は，並行処理可能であることを表している．異なるパスにはいる前には CPU 間の同期をとる必要があるが，同一パス内では，同期は不要である．なお，同図(a)，(b)の e, f で表されたレベル1から2への書込みは，論理的なページ場所であり，実際には b, c で読み込んだときのページに戻し，ページの呼びかえをする．このようにすることにより，CPU 間で e, f の書込みの同期がずれても，必要なページを上書きしてしまうことは避けられる．□

4.2 転送コストと台数の上限

本節では，前節のアルゴリズムの実行において，全処理を各装置に無駄なく分散させるための条件と方法について述べ，転送コストを見積もる．前節のアルゴリズムで無駄なく処理を分配できると，レベル 1-2 間について，レベル1の装置1台当りのデータ転送量は，4.1 節の F_1, F_2 を用いて，

$$f_1=F_1/u_1u_2$$
$$=p_2p_3't_1t_2/u_1u_2[\text{1次ページフェッチ}] \quad (10)$$

となる．同様にレベル 2-3 間について，レベル2の装置1台あたりのデータ転送量は，

$$f_2=F_2/u_2$$
$$=p_3't_2/u_2[\text{2次ページフェッチ}] \quad (11)$$

である．上式は，初期状態から最終状態への並べ替えが最もランダムな場合の転送コストになっている．そうでない場合は，レベル間の各パスにおけるページ参照について，次のパスのページにもなっている可能性があり，そのページの読込みを省略すると，式(10)，(11)よりさらに減少することができる．

まず，無駄なく処理を分配するための必要条件について考える．レベル 2-3 間については，レベル2にある全記憶容量 w_2u_2 [2次ページ] は，レベル3にある並べ替え対象となる全2次ページ数 p_3' を超えることはできない．そうでなければ，少なくとも $w_2u_2-p_3'$ [2次ページ] 分のレベル2の容量は使用されず無駄になってしまうからである．したがって，

$$u_2\leq p_3'/w_2 \quad (12)$$

が得られる．同様に，レベル 1-2 間について，1台のレベル2の装置に接続された u_1 台の装置の全記憶容量 w_1u_1[1次ページ] は，1台のレベル2の1次ページ数 w_2p_2 を超えることはできない．

$$u_1\leq w_2p_2/w_1 \quad (13)$$

したがって，レベル1，2の装置の台数の上限，$u_{1\max}$, $u_{2\max}$ は，

$$u_{1\max}=w_2p_2/w_1 \quad (14)$$
$$u_{2\max}=p_3'/w_2 \quad (15)$$

である．ここで，

$$a_1=u_{1\max}/u_1 \quad (16)$$
$$a_2=u_{2\max}/u_2 \quad (17)$$

とおくと，式(10)，(11)は次のようになる．

$$f_1=a_1a_2w_1t_1t_2$$
$$=a_1a_2w_1\lceil\log_{w_1}w_2\rceil\lceil\log_{w_2}(n/p_1p_2)\rceil \quad (18)$$
$$f_2=a_2w_2t_2=a_2w_2\lceil\log_{w_2}(n/p_1p_2)\rceil \quad (19)$$

ここで並べ替える全データ数（語数）が $n=p_1p_2p_3'$ の関係を用いている．これらの式から，u_1, u_2 を $u_{1\max}$, $u_{2\max}$ の定数分の1とすれば，a_1, a_2 は定数となり，nが十分大きいとして「 」をはずし近似すると，データ転送量は $\log n$ の定数倍となることがわかる．例えば p_1, p_2, w_1, w_2 が定数なら，$u_{1\max}$ は定数，$u_{2\max}$ は n に比例するので，上記のようにするには u_2 を n に比例して大きくする必要がある．

次に，十分条件について考える．レベル 2-3 間の u_2 台への処理の分配方法を具体的に述べることにする．$p_3'/u_2(=x_2)$ はレベル2の装置1台あたりの担当ページ数である．全パスを通じてレベル 2-3 間の処理は w_2 ページ単位にできる*ので，x_2 が w_2 の整数倍であれば，端数による無駄が生じない．x_2 の最小値は w_2 であり，このとき u_2 は最大値 $u_{2\max}$ をとる．$u_2=u_{2\max}/a_2(a_2=1,2,\cdots)$ のときは $x_2=a_2w_2$ となり，レベル2の各装置は a_2w_2 を担当する．したがって，a_2 が整数であれば，端数による無駄が生じない．このとき，レベル $2\#j(j=1,2,\cdots,u_2)$ は，level-3 にある第 $jx_2, jx_2+1,\cdots,(j+1)x_2-1$ ページを担当することになる．

レベル 1-2 間についても同様の議論ができる．$w_2p_2/u_1(=x_1)$ はレベル1の装置1台当りの担当ページ数である．x_1 が w_1 の整数倍であれば，端数による無駄が生じない．$u_1=u_{1\max}/a_1(a_1=1,2,\cdots)$ のとき，レベル1の各装置は a_1w_1 を担当する．このとき，レベル $1\#j(j=1,2,\cdots,u_1)$ は，level-2 にある第 jx_1, $jx_1+1,\cdots,(j+1)x_1-1$ ページを担当することになる．

CPU の延べ処理要素数は，$p_1p_2p_3't_1t_2$[語]，CPU 1台当り，$p_1p_2p_3't_1t_2/u_1u_2$[語] である．CPU の処理

* p_3' が w_2 のべき乗かつ行列データの転置の場合である．そうでない場合は端数が生じるが，影響は少ないので簡単のためこのようにした．

能力を T_0[語/s] としたので，処理時間は，$p_1p_2p_3't_1t_2/(u_1u_2T_0)$[s] である．データ転送ネックにしないためには，式(18)，(19)を時間に換算した，$f_1/T_1, f_2/T_2$ も同程度以下でなければならない．

[例2] 素朴法として，データを先頭から順に読みながら，別に用意されたレベル3の領域に読まれた順に目的ページに移動する方法を考える．全データ数は $p_1p_2p_3'$ である．レベル 2-3 間は，元のデータの読込みに p_3'[2次ページフェッチ] 必要である．各レベル2の2次ページ数が w_2 であることから，レベル 2-3 間のページヒット率を w_2/p_3' とすると，目的ページへ全データ $p_1p_2p_3'$ 個を移動するには，$p_1p_2p_3'(1-w_2/p_3')$ [2次ページフェッチ] 必要となる．したがって全体では，$F_2^N=p_3'+p_1p_2(p_3'-w_2)$ [2次ページフェッチ] となる．レベル 1-2 間については1次ページフェッチで勘定すると，同様にして $F_1=p_2p_3'+p_1(p_2-w_1)p_3'$ [1次ページフェッチ] となる．各レベルの装置間で処理が並列に動作可能とし，F_1^N, F_2^N を台数で割って，それぞれ $f_1^N=F_1^N/u_1u_2$, $f_2^N=F_2^N/u_2$ とする．$p_1=p_2=p_3'=100$, $w_1=w_2=10$, $u_1=u_2=10$ を例に比較する．この場合，$u_{1\max}=100$, $u_{2\max}=10$ から，式(16)，(17)は $a_1=10$, $a_2=1$ で整数となり，提案のアルゴリズムで無駄なく並列処理できる．これら数値を式(18)，(19)に代入すると，$f_1=200$, $f_2=20$ となるのに対し，素朴法では，$f_1^N=9,100$, $f_2^N=90,010$ である． □

4.3 データ転送能力

データ並べ替えの場合，必要とされるデータ転送能力を求めるため，3章で述べた，R_0, R_1 を用いて評価する．

CPU-キャッシュ間については，1要素ずつアクセスできるので，キャッシュに入れられた語については，1パスで並べ替えられる．したがって，レベル1からキャッシュに読み込まれた語は1回参照されるのみであるので，$R_0=1$ となり，式(4)から，

$$T_1=T_0/p_1\text{[1次ページ/s]} \tag{20}$$

が求まる．レベル 1-2 間については，パス数 t_1 回だけ同一ページが参照されるので $R_1=t_1$，したがって，

$$T_2=T_1u_1/p_2t_1\text{[2次ページ/s]} \tag{21}$$

となる．T_2 はレベル 1-2 間で要求されるデータ転送スピードを示すものである．データ転送量はパス数に比例するので，パス数は少ないほど効率がよい．$w_1=w_2$ のとき $t_1=1$ であり，式(21)から $T_2=T_1u_1/p_2$ とできることが望ましいが，この要求は現実的にはかなりきびしいものである．逆に，T_2 が十分大きくできない場合は，w_1 が大きくても無駄になることがわかる．4.4節に実現可能な T_2 が与えられたときに必要とされる w_1 の大きさを見積もっている．

4.4 各レベルの装置の記憶容量

4.1節にも述べたように，w_1 は最大 w_2 でよい．式(12)から，$w_2\le p_3'/u_2(=w_{2\max})$ であり，w_2 が $w_{2\max}$ より大きい場合，$w_2-w_{2\max}$ の部分は使用されず無駄になる．一方，w_1 については，式(21)から $t_1\equiv\lceil\log_{w_1}w_2\rceil=T_1u_1/T_2p_2$ であり，したがって，

$$w_1=w_2^{1/t_1}=w_2^{T_2p_2/T_1u_1} \tag{22}$$

に選べばよい．ただし，w_1 が整数でない場合は，整数に切り上げる．

レベル1，2の装置の記憶容量は，語数で表すと，

$$W_1=p_1w_1, \tag{23}$$

$$W_2=p_1p_2w_2 \tag{24}$$

となる．p_1, p_2 は，式(7)で転送スピードを確保するために必要な下限値が求まるので，w_1, w_2 の設定値から，W_1, W_2 に必要な容易が決まる．

5. m階層記憶モデル

m 階層記憶の場合には，レベル m にある p_m' 個の $(m-1)$ 次ページ分のレコードを並べ替えることになる．レベル i-$(i+1)$ 間 $(i=1,2,\cdots,m-1)$ をパス数 t_i で並べ替えるのに必要なデータ転送能力 T_i，レベル i に必要な記憶容量 w_i と W_i[語] は次のようになる．

$$T_1=T_0/p_1\text{[1次ページ/s]} \tag{25}$$

$$T_i=T_{i-1}u_{i-1}/p_it_{i-1}\quad(i=2,3,\cdots,m-1)\ [i\text{次ページ/s}] \tag{26}$$

$$w_i=w_{i+1}^{1/t_i}\quad(i=1,2,\cdots,m-2)\ [i\text{次ページ}] \tag{27}$$

$$w_{m-1}=p_m'^{1/t_{m-1}}\text{[}m-1\text{次ページ]} \tag{28}$$

$$W_i=p_1p_2\cdots p_iw_i\quad(i=1,2,\cdots,m-1)\ \text{[語]} \tag{29}$$

このとき，レベル i と $(i+1)$ $(i=1,2,\cdots,m-1)$ 間のデータ転送コスト f_i は次のようになる．

$$f_i=p_{i+1}\cdots p_{m-1}p_m't_i\cdots t_{m-1}/u_i\cdots u_{m-1}\ [i\text{次ページフェッチ}] \tag{30}$$

レベル i で無駄なく処理を分割させるための装置台数の上限 $u_{i\max}$ は次のようになる．

$$u_{i\max}=w_{i+1}p_{i+1}/w_i\quad(i=1,2,\cdots,m-2) \tag{31}$$

$$u_{(m-1)\max}=p_m'/w_{m-1} \tag{32}$$

$p'_m/u_{m-1}(=x_{m-1})$, $w_{i+1}p_{i+2}/u_i=(=x_i)$ $(i=1,2,\cdots,m-2)$ はレベル $j(j=1,2,\cdots,m-1)$ の装置1台当りの担当ページ数である．$a_j=x_j/w_j$ が整数であれば，端数による無駄が生じない．

現在，商用の並列計算機の記憶レベル数は2である[9),10)]．2階層記憶の場合は，上の結果で $m=2$ とすればよい．上式から，

$$T_1=T_0/p_1 \quad [1次ページ/s] \tag{33}$$

$$w_1=p_2'^{1/t_1} \quad [1次ページ] \tag{34}$$

$$W_1=p_1w_1 \quad [語] \tag{35}$$

$$f_1=p_2't_1/u_1 \quad [1次ページフェッチ] \tag{36}$$

$$u_{1\max}=p_2'/w_1 \tag{37}$$

となる．式(36)は，パス数 $t_1=\lceil\log_{w_1} p_2'\rceil$ のアルゴリズムを，CPU 台数 u_1 で並列処理することを表している．$1/p_1$ は単位の変換係数なので，式(33)からは，データ参照の局所性がなく計算処理能力 T_0 と等しいデータ転送能力を要求していることがわかる．

[例3] 具体的な並列計算機のパラメータにより式(33)～(37)を見積もってみる．富士通 VPP 500[9)]の場合，本論文での2階層記憶モデルで，レベル2のメモリを省略しレベル1のメモリを直結した場合に相当する．PE 台数は最大構成で $u_1=222$ である．p_2' は並べ替えの対象となるデータの量で決まる．例えば，$p_2'=8,000$ をパス数 $t_1=3$ で並べ替える場合，$w_1=20$, $W_1=20p_1$, $f_1=108$, $u_{1\max}=400$ になる．1[語]=8[Byte] とし，PE の処理能力あるいはローカルメモリ-PE 間のデータ供給能力 T_0 が 800 [M語/s] であるとすると，式(33)からレベル1-2間のデータ転送能力は $T_1=800/p_1$[M1次ページ/s]=800[M語/s] を要求される．一方，VPP 500 の公表データによるとレベル 1-2 間のデータ転送能力は 50[M語/s] なので，上記 T_1 の 1/16 にすぎない．すなわち，データ並べ替え処理を行う限りにおいて，データ転送ネックになり，PE の処理能力は十分には生かされない． □

[例4] Dash[8)]の場合も 1[語]=8[Byte] とし，他のパラメータを $u_1=4$, $u_2=16$, $T_0=8$[M語/s], $T_1=2$[M語/s]*, $T_2=30$[M語/s]** とする．一方，レベル1-2 間の要求される通信容量は，$T_1=8$[M語/s] であり，現実値はその 1/4 である．レベル2間については，$t_1=1$ としても要求される通信容量は 32[M語/s] であり，現実値はそれにほぼ等しい．すなわち，レベル1-2間は転送ネック，レベル2間はほぼ通信容量が確保されていると見ることができる．なお，先の VPP 500 とは PE の処理性能が大幅に異なるためこれらの結果を一概には比較できない． □

* レベル1-2間はバス結合のため u_1 台で時分割使用するものとし，バスの通信容量を $1/u_1$ 倍した．

** レベル2の装置間は2次元のルーティングネットワークなので，レベル2全体の通信容量は一経路の通信容量 7.5 [M語/s] を $\sqrt{u_2}$ 倍した．

6. むすび

共有メモリをもつマルチ CPU マシンをモデル化し，データ転送ネックにならないために必要な転送能力を与えた．具体的適用例としてデータ並べ替え問題に的を絞り，処理時間が単一 CPU マシンの場合の下界に対して，CPU 台数分の1でできる並列アルゴリズムを示した．また，細粒度並列処理への適用を考慮し，並列処理可能な各レベルの装置台数の上限を与え，データ数が n で，n に比例する台数の装置を用いる場合，計算時間およびデータ転送量が高々 $\log n$ の定数倍であることを示した．さらに，CPU の計算能力を最大限に発揮させるために必要な，レベル間のデータ転送能力，各レベルの記憶容量を求めた．

現在，商用の並列計算機の記憶階層数は2であるが，プロトタイプ段階のもので，本論文の階層モデルで階層数3の並列計算機が出現している．本論文の結果は，現存する2レベルの記憶階層をもつ並列計算機だけでなく，今後回路の集積化が進み，出現が予想される3レベル以上の記憶階層をもつ超並列計算機についても，そのレベル間に要求されるデータ転送能力を見積もるものである．

参考文献

1) Babb, R. G.: 並列処理マシン用 Fortran の「HPF」業界標準の地位を目指す，日経エレクトロニクス，No. 581, pp. 187-199 (1993).
2) Bailey, D. H.: Experience with Parallel Computers at NASA Ames, *RNR Technical Report*, RNR-91-007, pp. 1-11 (1991).
3) Floyd, R. W.: Permuting Information in Idealized Two-Level Storage, *Complexity of Computer Computations*, Miller, R. and Thatcher, J. (ed.), Plenum Press, NY, pp. 105-109 (1972).
4) 佐藤，津田：2階層記憶における効率のよいデータ並びかえアルゴリズム，情報処理学会論文誌，Vol. 27, No. 9, pp. 845-852 (1986).
5) 佐藤，津田：多階層記憶における効率のよいデータ並べかえと記憶階層の最適化，情報処理学会論文誌，Vol. 29, No. 4, pp. 386-396 (1988).
6) Aggarwal, A. and Vitter, J. S.: The Input/Output Complexity of Sorting and Related Problems, *Comm. ACM*, Vol. 31, No. 9, pp. 1116-1127 (1988).

7) Imadeldin, O. M. and Elmagarmid, A. K.: Performance Analysis of a Generalized Class of *m*-Level Hierarchical Multiprocessor Systems, *IEEE Trans. Parallel and Distributed Systems*, Vol. 3, No. 2, pp. 129-138 (1992).
8) Lenoski, D. et al.: The Stanford Dash Multiprocessor, *IEEE Computer*, Vol. 25, No. 3, pp. 63-79 (1992).
9) 浅見，宮崎：355.2 GFLOPS のスーパーコンピュータを富士通が開発，日経エレクトロニクス，No. 564, pp. 100-102 (1992).
10) 稲葉，安保：4プロセサ構成で 32 GFLOPS のスーパーコンを日立が発売，日経エレクトロニクス，No. 551, pp. 88-89 (1992).

(平成5年9月 8 日受付)
(平成6年1月13日採録)

佐藤 隆士（正会員）

昭和28年9月生．昭和51年岡山大学工学部電子工学科卒業．昭和53年同大学院修士課程修了．同年詫間電波工業高等専門学校助手．現在大阪教育大学教養学科情報科学専攻助教授．工学博士．記憶階層のアルゴリズム，並列処理，データベース，文字列検索アルゴリズムの研究に従事．電子情報通信学会，CAI学会，IEEE Computer Society，ACM 各会員．

津田 孝夫（正会員）

1957年3月，京都大学工学部電気工学科卒業．現在京都大学工学部情報工学科教授．計算機ソフトウェア講座担当．工学博士．自動ベクトル化/並列化コンパイラ，スーパコンピューティング，オペレーティングシステムの研究に従事．「モンテカルロ法とシミュレーション」（培風館），「現代オペレーティングシステムの基礎」（オーム社，共著」，「数値処理プログラミング」（岩波書店）などの著書がある．昭和63年度および平成3年度情報処理学会論文賞受賞．元本学会関西支部長．ACM. SIAM 各会員．IFIP/WC 2.5 (Numerical Software) 委員．