仮想メモリ・システムの二次記憶管理の最適化†

西垣 通†† 緒方慎八†††
大町一彦†† 池田智明††

仮想メモリ・システムの主記憶管理に関する報告は従来より多くなされているが，二次記憶管理に関する報告は知られていない．仮想空間上のページと二次記憶上のスロットとを対応づけるスロット割り当て方式により，ページ入出力実行時間は異なり，システムの性能は影響をうける．従来方式として固定方式と浮動方式とが代表的だが，両者の優劣はプログラム特性に依存する．

本論文は，プログラム特性によらず従来の二方式より効率のよいスロット割り当て方式の実現に関する．まず，従来方式を包含した解析モデルを設定し，ページ入出力実行時間の最小化をもって最適性を定義する．次に，実用的条件下で最適解を与える方式として，「集中方式」を新しく提案する．さらに，本方式を実現し，実測評価を行い，ページ入出力実行時間が従来方式より短縮された結果を示す．

なお，本方式はすでに実用化されている．

1. まえがき

仮想メモリ方式の計算機システムの問題点は性能にあると指摘されている．仮想メモリ方式のもとでは，仮想空間上のプログラムやデータはページ単位に分割されてドラム，ディスク等の二次記憶媒体に蓄積され，参照の必要に応じて主記憶に読みこまれる．したがって性能向上のためには，第1に参照確率の高いページの主記憶内保持によるページ入出力実行回数の削減，第2に二次記憶媒体上のページ蓄積法の適正化によるページ入出力実行時間の短縮，の2点が必要となる．前者は従来より研究され[4)~7)]ており，多重プログラミングのもとでの性能もDenning[6)]，Masuda[7)]らにより論じられた．しかし後者を論じた報告は，筆者の知る限り見当たらない．ドラム，ディスクの入出力効率向上に関する報告はあるが[2),3)]，仮想メモリの二次記憶媒体としての取扱いはなされていない．ページ入出力は，スワッピングなど特有の実行要求特性をもち，その実行時間はページが蓄積された二次記憶上のスロットの位置に依存する．したがってスロット割り当ての適正化により性能改善を達成しうる．

従来用いられるスロット割り当て方式として，ジョブの開始から終了まで各ページに常に一定のスロットを対応させる固定方式[11)]と，ページ書き出し時点でその書き出し時間を最小化する位置のスロットを割り当てる浮動方式[12)]とが代表的である．例えばIBM社のオペレーティング・システムVS 1, VS 2では各々前者，後者が採用されている．特に二次記憶媒体がディスクの場合，あるページが書き出されるスロットは，固定方式では常に同一だが，浮動方式ではたまたまディスク・ヘッドが停止していたシリンダから選ばれる．したがって，ページ読みこみの際のシーク時間も大きく異なる．筆者は，多重プログラミングのもとでの両方式のページ入出力効率を解析的に比較した[1)]．この結果，優劣はプログラム特性に依存し，スワップ・インされた後スワップ・アウトされるまでにページに書きこみが行われる確率が大の場合は浮動方式，小の場合は固定方式が各々優れることが判明した．

本論文は，プログラム特性によらず従来方式より優れたページ入出力効率を実現する方式に関する．ジョブの実行を通じて，あるページに割り当てられるスロットの存在しうる範囲に着目すると，固定方式では二次記憶上の一点であるが，浮動方式ではその全域に等しい．割り当ての自由度が増しこの範囲が広がるにしたがって，書き出し時間は減少する．一方，スワップ・イン時のアクセス範囲は拡大するので読みこみ時間は増大の傾向をもつ．したがって両者の和に着目すれば，最適な範囲ないし自由度が存在すると考えられる．第2, 3章では，この自由度／範囲をパラメータとしたモデルを設定し，自由度最小，最大の場合を各々固定方式，浮動方式に対応させる．ページ入出力実行

† An Optimized Secondary Storage Management in a Virtual Memory System by TOHRU NISHIGAKI (Systems Development Laboratory, Hitachi, Ltd.), SHINPACHI OGATA (Software Works, Hitachi, Ltd.), KAZUHIKO OHMACHI and CHIAKI IKEDA (Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所
††† (株)日立製作所ソフトウェア工場

時間の最小化をもって本モデルにおける最適性を定義し，最適解を求める．さらに実用的条件下で最適解を与える方式として「集中方式」を新しく提案する．一括して入出力実行されるページの対応スロット群の位置が比較的集中化される点が，本方式の特徴である．二次記憶媒体として可動ヘッドのディスクを想定すると，本方式における自由度／範囲は1シリンダに等しくなる．第4章では，本方式を実現し，その性能を従来方式と比較測定した結果を提示する．

2. 二次記憶のスロット割り当て

まず本論文で用いる用語および前提条件を整理する．「スロット」とは，二次記憶媒体上のページ単位に区分されたレコードである．「スワッピング」とは，ジョブの working-set[5] を二次記憶と主記憶との間でやりとりする行為を表わす．

基本的な前提条件は以下の通りである．

- 二次記憶媒体としては，スロット位置が入出力効率に与える影響が大であることから可動ヘッドのディスク[10]を仮定する．
- 主記憶管理方式は working-set scheduler[5] を仮定する．working-set はスワップ・イン時に一括して読みこまれ (pre-paging)，スワップ・アウト時に一括して書き出される．
- アプリケーションとしては，そのスワッピングが二次記憶媒体の負荷の大半を占める[8] と予想されるインタラクティブな端末起動ジョブを想定し，各ジョブは多重プログラミングで処理されるとする．なおジョブ間でのページ共用の影響は無視する．
- スペース利用率とページ入出力効率の関係は考察の範囲外とし，ページ書き出し時には常に必要なだけの未割り当てスロットが存在すると仮定する．

次にスロット割り当て方式の効率を比較するための評価関数を設定する．ジョブはスワップ・インされて実行され，しかるのちスワップ・アウトされる．この間にページ・フォールト (working-set へ加わるページの読みこみ) とページ・アウト (working-set から外れ，内容が変化したページの書き出し) とが実行される．スワップ・イン→ページ・フォールト／アウト→スワップ・アウトは1つのサイクルをなし，インタラクティブな端末起動ジョブのページ入出力は，比較的短小なサイクルの反復より生ずる．本論文では，1ジョブの1サイクルにともなうページ入出力実行時間を評価関数とし，式(1)で定義する．

$$C \triangleq n_{SO}\cdot C_{SO}+n_{SI}\cdot C_{SI}+n_{PO}\cdot C_{PO}+n_{PF}\cdot C_{PF} \quad (1)$$

$C_{SO}, C_{SI}, C_{PO}, C_{PF}$ はそれぞれ swap-out cost, swap-in cost, page-out cost, page-fault cost であり，各行為1回あたりの平均アクセス時間を表わす（ページ入出力実行時間は転送時間とアクセス時間との和であるが，前者はスロット割り当て方式によらないので，以下後者のみに着目する）．n_{SO}〜n_{PF} は1サイクルあたりの各行為の頻度を表わす．ここで定義より $n_{SO}=n_{SI}=1$．また，ページ・アウトは実際には working-set から外れた直後でなくある程度まとめて行われ[1]，かつ1サイクルが比較的短いことから，$n_{PO}=1$ と仮定する．この仮定については **3.3** でふれる．さらに式(1)の第4項は，スロット割り当て方式によらないので除外できる．これは第1に，多重プログラミングのもとでページ・フォールトが発生したとき，当ページに対応するスロットの位置ならびにディスク・ヘッドの位置は，ともにランダムと考えられること，第2にページ・フォールト発生率もプログラム特性より定まりスロット割り当て方式には依存しないためである．以上より，式(1)をさらに次のように再定義する．

$$\begin{cases} C \triangleq C_{SO}+C_{SI}+C_{PO} \\ C_{SO} \triangleq f_{SO}+g_{SO} \\ C_{SI} \triangleq f_{SI}+g_{SI} \\ C_{PO} \triangleq f_{PO}+g_{PO} \end{cases} \quad (2)$$

f, g は各々シーク時間，サーチ時間の期待値を表わす．C を total cost とよび，これを最小化するスロット割り当て方式を最適方式とする．

上記評価関数に関し，従来方式の得失は次のとおりである．固定方式では，各ページに割り当てられるスロットはジョブの実行開始から終了まで不変であり，1つのジョブのプログラムのページ群に対応するスロット群は1シリンダないし隣接した数シリンダ上に存在する．したがって，アクセスするシリンダの位置，シリンダ内のスロットの位置，ディスク・ヘッドの位置が各々一様分布にしたがうと仮定すれば，f_{SO}, f_{SI}, f_{PO} はすべてランダム・シーク1回に要する時間の期待値となり，g_{SO}, g_{SI}, g_{PO} は各々，ランダム・サーチ時間の期待値に入出力実行するページ数を乗じた値となる．

浮動方式では，ジョブ実行中に内容が変更されたページは，書き出し時点でディスク・ヘッドが停止しているシリンダ内の連続したスロット群に書き出される．この場合，f_{SO}, f_{PO} は無視でき，g_{SO}, g_{PO} は1

回のランダム・サーチ時間期待値で近似できる．一方 f_{SI} は，working-set 内のページに対応するスロット群が存在するすべてのシリンダをスキャンするシーク時間の期待値[1]である．また g_{SI} は，上記スロット群がディスク上で形成する相異なるブロック数[1]にランダム・サーチ時間期待値を乗じた値となる．(ただし以下，スワップ・アウト／ページ・アウト時に一括して書き出されるページ群に割り当てられるスロット群を「ブロック」とよぶ．1ブロック中のスロット群は，ディスク上で物理的に連続しているとみなす．)

両方式を比較すると，C_{SO} と C_{PO} は浮動方式が，C_{SI} は固定方式が，各々小さい可能性が高く，Cの大小はプログラム特性に依存する[1]．

3. スロット割り当ての最適化

3.1 スロット割り当てモデル

従来方式を包含するスロット割り当ての一般モデルを提示する．なお以下，二次記憶用の領域はディスク上で連続したLシリンダを占め，ジョブのプログラムは各々1シリンダに収納可能と仮定する．

スワップ・アウト／ページ・アウトの際，ディスク・ヘッドが第 $(a+1)$〜$(a+h)$ シリンダ (**図1**の斜線部) にあれば，これが現在停止しているシリンダ内の連続したスロット群（ブロック）に書き出す．ディスク・ヘッドが第1〜aシリンダにあれば第 $(a+1)$ シリンダ内（ただし $a\neq 0$ のとき），第 $(a+h+1)$〜L シリンダにあれば第 $(a+h)$ シリンダ内（ただし $a+h\neq L$ のとき）の，各々連続したスロット群(ブロック)に書き出す．ここでhは全ジョブで共通だが，aはジョブごとに異なる値をとる．すなわち本モデルで，hはスロット割り当ての自由度を表わし，各ジョブに割り当てられるスロット群は，ジョブごとに連続したhシリンダ内にまとまる．hの値域は式（3）で与えられ，hが定められたときaの値域は式(4)で与えられる．

$$1\leq h\leq L. \tag{3}$$

$$0\leq a\leq L-h. \tag{4}$$

hの増大にともない f_{SO} と f_{PO} は減少するが f_{SI} は増大するため，Cを最小にするhの最適値が存在する．なお g_{SO}, g_{SI}, g_{PO} はhやaの値によらない．

本モデルで $h=L$ とおくと，浮動方式と一致する．また本モデルで $h=1$ とおくと，これは固定方式に近いが，より自由度が大であり，以下の理由により固定方式より total cost C が小さい．まず，両者ともシリンダは固定なので f_{SO}, f_{SI}, f_{PO} は同一である．次に，本モデル ($h=1$) ではシリンダ内のスロット割り当ては浮動でありスワップ・アウト／ページ・アウトの際連続したスロット群に書き出すが，固定方式ではこれも固定であり書き出すスロット群は一般に非連続なので，g_{SO}, g_{PO} は前者の方が小さい．さらに，working-set 内のページが属する相異なるブロック数は当然 working-set size 以下であり，g_{SI} は各々これにランダム・サーチ時間期待値を乗じた値であるから，g_{SI} も前者の方が小さい*．

以上より，本モデルでhの最適解を求めると，それは浮動方式，固定方式のいずれよりも効率がよい．

3.2 パラメータ h の最適化

本モデルにおいて式(1)(2)の諸元を導く．なお以下，ディスクのシーク時間を**図2**に示す傾きKの一次関数で近似する．S_0 は最小シーク時間である[10]．また，シーク動作は，スキャン方式**にしたがうとす

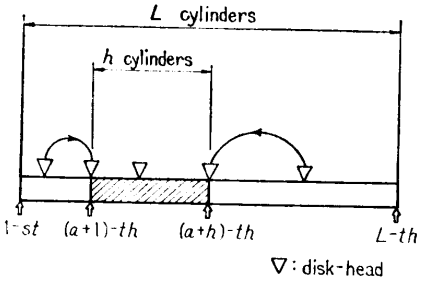

図 1 スロット割り当てモデル：スワップ・アウト／ページ・アウト時のシーク動作
Fig. 1 The slots allocation model: the seek operation at swap-out/page-out.

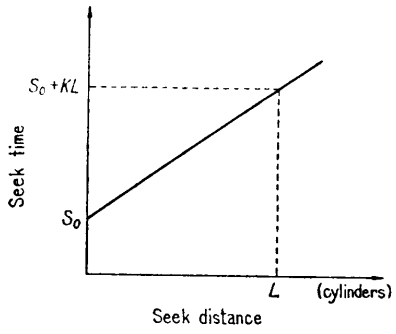

図 2 ディスク・シーク時間の一次関数による近次
Fig. 2 A continuous approximation for disk seek time.

* 一般に $h=1$ より自由度を減ずるとCは増大することが，同様にしていえる．

** 入出力要求をシリンダ番号の若い順にソートし，ディスク・ヘッドのスキャンにより順次アクセスする方式．

る．1)～3)サーチ動作は特にソートせずランダム・サーチを行うとする．さらに，ページ入出力のためのシーク動作開始時点でのディスク・ヘッド位置は1～Lの一様分布にしたがうと仮定する．

$$f_{SO}(h,a)=f_{PO}(h,a)=(a/L)(S_0+Ka/2)+\{(L-a-h)/L\}\{S_0+K(L-a-h)/2\}. \quad (5)$$

式(5)で，第1項の(a/L)はディスク・ヘッドがシーク動作開始時点で第1～aシリンダにある確率であり，$(S_0+Ka/2)$はそのときの期待シーク時間を表わす．第2項はディスク・ヘッドが第$(a+h+1)$～Lシリンダにある場合につき同様に求めたものである．

次に$f_{SI}(h,a)$を導く．working-set内のページに対応するスロット群がx個の相異なるブロックに属するとする（xブロックに断片化する理由については，1)を参照されたい）．これらのブロックをスワップ・アウト／ページ・アウトで形成するためのシーク動作開始時点における，相異なるディスク・ヘッド位置の数の期待値をyとする（ただしディスク・ヘッドが図1の斜線部にあれば，実際にはシークは行われない）．一様分布の仮定のもとで，yはxより算出できる[9]．$L\gg x$のときは$y\cong x$であるが，xの増大につれて$y<x$となる（付録図10参照）．$h=L$のとき，x個のブロックが第1～Lシリンダの範囲のy個のシリンダにそのまま散在する．一般の場合には，x個のブロックが存在するシリンダは第$(a+1)$～$(a+h)$シリンダの範囲に限られ，これらをアクセスするための期待シーク時間は次式で与えられる．

$$f_{SI}(h,a)=[\{(L-1)/L\}S_0+(a/L)(Ka/2)+\{(L-a-1)/L\}\{K(L-a-1)/2\}]+\{y(h-1)/L\}\{S_0+KL/(y+1)\}. \quad (6)$$

式(6)の第1項は，スキャンの準備動作としてディスク・ヘッドが第$(a+1)$シリンダに到着するための期待シーク時間である．a/Lはシーク開始時点でディスク・ヘッドが第1～aシリンダ，$(L-a-1)/L$はこれが第$(a+2)$～Lシリンダにある確率であり，$a/2$，$(L-a-1)/2$は各々の場合の期待シーク距離を表わす．第2項は第$(a+2)$～$(a+h)$シリンダをスキャンするための期待シーク時間であり，$y(h-1)/L$はこの範囲でアクセスすべきシリンダ数の期待値である．ここで，1シリンダあたりの期待シーク距離が$L/(y+1)$である[2),3)]ことに注意されたい．

期待サーチ時間g_{SO}, g_{SI}, g_{PO}は次式で与えられる．

$$g_{SO}=g_{PO}=R/2. \quad (7)$$

$$g_{SI}=x\cdot R/2. \quad (8)$$

ただしRはディスクの回転時間を表わす．

式(5)～(8)を式(2)に代入すると，Cがaとhの関数として求まる．ここで，多重プログラミングで処理されるジョブの数が多ければ，aは式(4)の範囲で一様分布にしたがうとみなせる．したがって，平均としての total cost $\bar{C}(h)$ を次式で定義する*．

$$\bar{C}(h)\triangleq\{1/(L-h)\}\cdot\int_0^{L-h}C(h,a)da$$
$$=K(L-h)^2/L+\{2S_0-K(L-1)/2\}(L-h)/L+\{S_0+KL/(y+1)\}y(h-1)/L+K(L-1)^2/(2L)+(L-1)S_0/L+(x+2)R/2. \quad (9)$$

式(9)をhで微分すると次式をうる．

$$\frac{d\bar{C}}{dh}=-2K(L-h)/L-\{2S_0-K(L-1)/2\}/L+\{S_0+KL/(y+1)\}y/L. \quad (10)$$

これより，

$$\frac{d\bar{C}}{dh}=0, \quad (11)$$

をみたす$\hat{h}$は，次式で与えられる．

$$\hat{h}=-Ly/\{2(y+1)\}-S_0(y-2)/(2K)+(1/4+3L/4). \quad (12)$$

$\bar{C}(h)$は，hの下に凸な二次曲線であるから，式(3)の条件より，$\hat{h}\leq 1$なら$h=1$，$\hat{h}\geq L$なら$h=L$，それ以外のとき$\hat{h}$が最適解を与える．

3.3 集中方式 (Concentrating Technique)

実際には，実用的条件下で$\hat{h}\leq 1$が成立し，$h=1$を近似的な最適解とみなしうる．以下$h=1$を「集中方式 (Concentrating technique) とよぶ，集中方式では，シリンダ割り当ては固定（常に第$(a+1)$シリンダに書き出す），シリンダ内のスロット割り当ては浮動であり，その実現は比較的容易である．

式(12)より，$\hat{h}\leq 1$の条件は次式で与えられる．

$$y\geq(1/S_0)\cdot\{(S_0/2+KL/4-3K/4)+\sqrt{\{S_0/2+KL/4-3K/4\}^2+2S_0\{S_0+3KL/4-3K/4\}}\}. \quad (13)$$

HITAC H-8589-1 ($S_0=10$ ms, $K=0.1125$ ms/シリンダ) および H-8589-11 ($S_0=10$ ms, $K=0.05625$ ms/シリンダ) の2種のディスク[10]について，式(13)の条件を図示すると**図3**の斜線部のようになる（なお，Lには下限値が存在するので，ここでは$L\geq 10$として示した）．実際にはyが斜線部内にある場合が多いと考えられる[1)]．また，

* hやaは整数値だが，連続値をとると仮定して議論を進める．

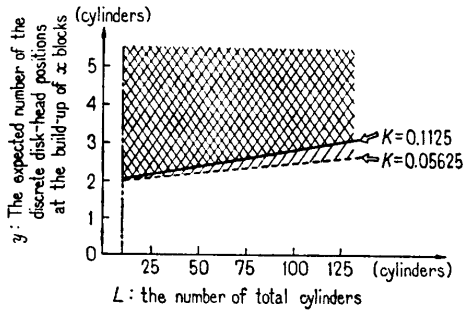

図 3 集中方式が最適となる条件 $\hat{h}\leq 1$ (斜線部)
Fig. 3 The condition yielding $\hat{h}\leq 1$(shaded portion), where the Concentrating technique is optimum.

$$\frac{d\bar{C}}{dy}=(h-1)S_0/L+K(h-1)/(y+1)^2 + \frac{dy}{dx}\cdot(R/2)>0, \tag{14}$$

であり（付録より $dy/dx>0$），y の減少にともなって $\bar{C}$ も減少する．したがって y が小さい場合（斜線部以外）には $\bar{C}$ 自体が小さいので，実用上余り問題とならない．

集中方式は，最適解とならない条件のもとでも，多くの場合従来方式よりも優れている．まず固定方式と比較すると，すでに **3.1** で示したように，集中方式の total cost は L, x, y の値によらず常に固定方式のそれより小である．したがって本論文では，浮動方式との比較を中心に述べる．

以下，集中方式 (Concentrating : CN) と浮動方式 (Floating : FL) のコストを右肩の記号で区別し，$C^{CN}<C^{FL}$ の成立する条件を求める．C^{CN}, C^{FL} は各各式(9)より次のように与えられる．

$$C^{CN}=\bar{C}(1)=K(L-1)^2/L+3S_0(L-1)/L+(x+2)R/2. \tag{15}$$

$$C^{FL}=\bar{C}(L)=\{S_0+KL/(y+1)\}y(L-1)/L+K(L-1)^2/2L+S_0(L-1)/L+(x+2)R/2. \tag{16}$$

式(15)，(16)より $C^{CN}<C^{FL}$ の成立する条件は，

$$y>(1/S_0)\cdot[(S_0/2-KL/4-K/4)+\sqrt{\{S_0/2-KL/4-K/4\}^2+2S_0\{S_0+KL/4-K/4\}}], \tag{17}$$

で与えられ，$S_0=10$ ms ; $K=0.1125$, 0.05625 ms/シリンダの場合につき，この範囲を斜線部で示したのが**図 4** である（図 3 と同じく $L\geq 10$ とした）．図 4 の斜線部は，現実に生じうる場合をほぼつくしていると考えられる[1]．

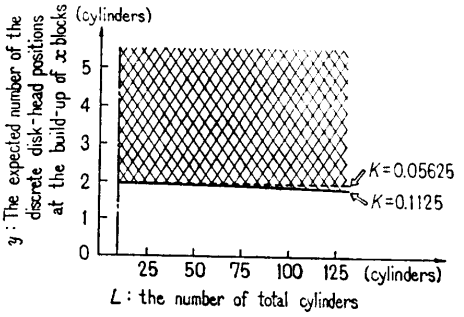

図 4 集中方式が浮動方式より優れる条件 $C^{CN}<C^{FL}$ (斜線部)
Fig. 4 The condition yielding $C^{CN}<C^{FL}$ (shaded portion), where the Concentrating technique is superior to the Floating technique.

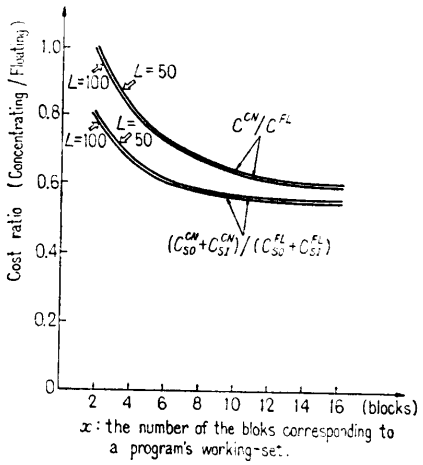

図 5 集中方式と浮動方式とのコスト比較
Fig. 5 The paging cost of the Concentrating technique in comparison with the Floating technique.

さらに，$S_0=10$ ms ; $K=0.05625$ ms/シリンダ (H-8589-11) ; $L=50$, 100 シリンダの条件のもとで，両方式のコスト比 $(C_{SO}^{CN}+C_{SI}^{CN})/(C_{SO}^{FL}+C_{SI}^{FL})$, C^{CN}/C^{FL} をブロック数 x の関数として示したのが**図 5** である．ただしここで，swap-out cost C_{SO}^{CN}, C_{SO}^{FL} および swap-in cost C_{SI}^{CN}, C_{SI}^{FL} はそれぞれ次のように求められる (page-out cost は swap-out cost に等しい)．

$$C_{SO}^{CN}=\{1/(L-1)\}\cdot\int_0^{L-1}C_{SO}(1,a)da =K(L-1)^2/(3L)+S_0(L-1)/L+R/2. \tag{18}$$

$$C_{SO}^{FL}=C_{SO}(L,0)=R/2. \tag{19}$$

$$C_{SI}^{CN}=\{1/(L-1)\}\cdot\int_0^{L-1}C_{SI}(1,a)da$$

$$=K(L-1)^2/(3L)+S_0(L-1)/L+xR/2. \quad (20)$$

$$C_{SI}^{FL}=C_{SI}(L,0)=\{S_0+KL/(y+1)\}y(L-1)/L+K(L-1)^2/2L+S_0(L-1)/L+xR/2. \quad (21)$$

xについては，次章で実測データにもとずく推定値を提示する．また，working-set size, ページに書きこみの行われる確率，ページ・フォールト発生率よりxを推定する議論については，1) を参照されたい．

図5より，集中方式を浮動方式と比較すると，total cost として約 0～40% の効率向上が期待される．実際の効率向上効果は，ページ・アウトの頻度がスワップ・アウト／インの頻度と同程度ならばC^{CN}/C^{FL}で評価でき，これが減少するにしたがって$(C_{SO}^{CN}+C_{SI}^{CN})/(C_{SO}^{FL}+C_{SI}^{FL})$に近づくと考えられる．

なお，図5で $L=50$ と 100 とを比べると，$L=50$ がやや効率向上効果が小さいが，両者の差はわずかである．また，$K=0.1125$ ms/シリンダ (H-8589-1) の場合は，識別できない程度に図5と一致した．

4. 実験と効果測定

4.1 測定条件と測定方法

集中方式を実験的に実現し，その効果を浮動方式との比較において検証した結果を以下に示す．初めに測定の条件と方法をのべる．

実測評価の機器構成としては，当社の大型計算機 HITAC M-180 を使用し，二次記憶媒体は H-8589-11 ディスク[10]を用いた．負荷としては，短小な入力編集処理を実行する 50 端末の稼動中を仮定した．50 人の測定人員を確保することは困難なので，端末の回線入出力をディスクへの入出力により模擬する「端末シミュレータ」を用いて測定を行った．端末シミュレータは，各端末ごとに，平均 20 秒の指数分布にしたがう思考時間が終了すると入力編集コマンドを発生して処理を要求する．回線入出力模擬用のディスクと二次記憶用ディスクは別の媒体を使用し，また回線入出力以外の動作は全く実システムの動作と一致している．したがって，二次記憶管理の方式評価に関する限り，実際に 50 人が端末使用中と同様の周囲条件にあるとみなすことができる．なお，バッチ・ジョブは実行しなかった．

* スワップ・イン中の端末は，スロット割り当て実行中なので除外した．

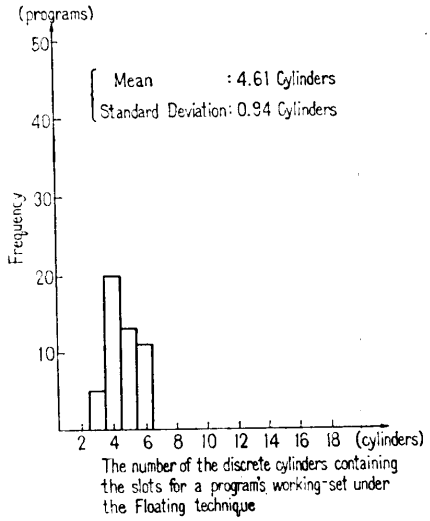

図 6 浮動方式のもとで求めた y の推定値
Fig. 6 The distribution used for the estimation of y, observed at a steady state under the Floating technique.

測定は2つの方法により行った．第1は実メモリのダンプである．これによりyの推定値を求めた．既述のように，浮動方式のもとでは，yは working-set 内のページに割り当てられたスロット群が存在する相異なるシリンダ数に等しい．実メモリ内の管理テーブルのダンプ情報から，この推定値を算出した．第2は本実験のために開発したページ入出力効率評価用のソフトウェア・モニタである．本モニタはページ入出力が実行される度に，その実行に要した時間を記録し，逐次磁気テープに出力するものである．

なお，測定時間は各方式につき約30分間とし，全50端末が log-on を完了し入力編集コマンドを発生中の定常状態約16分間について測定データを整理した．

4.2 測定結果

浮動方式のもとで定常状態になってから約10分間経過した時点で，yの推定値を測定した結果が**図6**である．図6は，この時点でたまたまスワップ・アウトされていた*49端末について，各々スワップ・アウト動作直前における working-set 内のページに割り当てられたスロット群の存在する相異なるシリンダ数の度数分布を表わす．平均値は 4.61 シリンダ，標準偏差は 0.94 シリンダである．**図7**はこのときの working-set size の度数分布を示す．平均値は 9.18 ページ，標準偏差 は1.24 ページである．したがって約2ページずつ各シリンダに散在していることがわかる．

集中方式および浮動方式のもとで，定常状態約16分

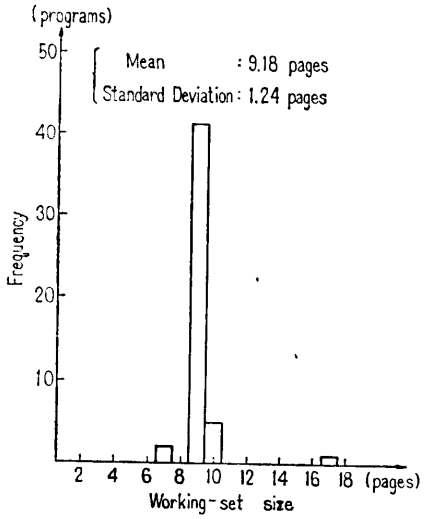

図 7 浮動方式のもとでの working-set size の度数分布
Fig. 7 The distribution of the observed working-set size at a steady state under the Floating technique.

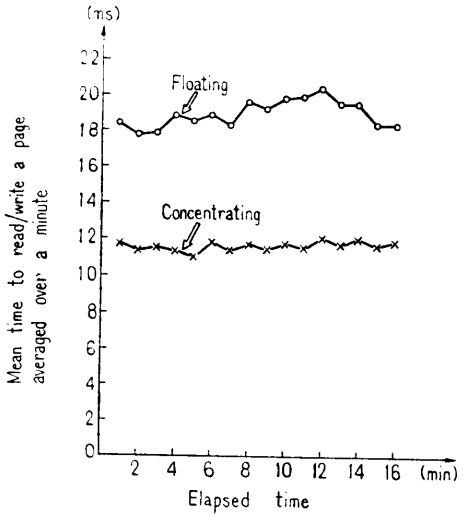

図 9 ページ入出力実行時間平均値の時間変化
Fig. 9 The mean page I/O time versus the elapsed time.

間にわたって測定したページ入出力実行時間の度数分布を**図 8** に示す．集中方式では，ページ入出力実行時間平均が浮動方式より約 38% (18.88 ms から 11.63 ms) 短縮された．また，同標準偏差，同 90% 累積値においても顕著な短縮が見られた．なお，入出力ページ数の統計は 31,973 ページ (集中方式)，32,008 ページ (浮動方式) であり，その契機の内訳はスワップ・アウト，スワップ・イン，ページ・アウト，ページ・インが各々 43.97%，52.93%，0.10%，3.00% (集中方式)，44.04%，52.92%，0.13%，2.91% (浮動方式) であった．

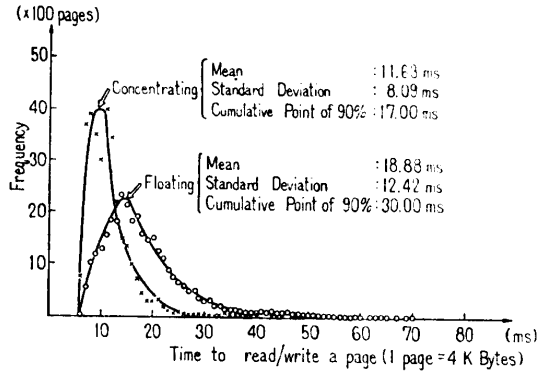

図 8 集中方式 (×) と浮動方式 (○) とのページ入出力実行時間の実測比較
Fig. 8 The distributions of the observed page I/O time for the two techniques, Concentrating (×) and Floating (○).

図 5～8 により，期待された効果の検証を行うことができる．図 6 に示したシリンダ数平均値は一時点におけるポイント・データに過ぎず，測定全体を通じての統計値ではない．しかし，**図 9** に示すように，1 分ごとのページ入出力実行時間平均値の時間変化が比較的小さいことから，一応 y の推定値として採用可能である．ここで，y が約 4.6 のとき x は約 4.8 である (**図 10 参照**)．ページ入出力実行時間平均値と total cost C とは直接対応はしないが，集中方式と浮動方式とのページ入出力実行時間平均値の比率 0.62 は図 5 の $x=4.8$ における $(C_{SO}^{CN}+C_{SI}^{CF})/(C_{SO}^{FL}+C_{SI}^{FL})$ の値に比較的よく一致している．いまページ・アウトの頻度が非常に低く，既述のようにページ入出力効率は swap-out cost と swap-in cost の和で評価できることから，ほぼ期待された効果を実現することができたと考えられる．

5. むすび

仮想メモリの二次記憶と主記憶との間のページ入出力効率向上を目的として，「集中方式」とよぶ新しい二次記憶管理方式を提案した．まず，従来方式を含め各方式の効率を比較するための解析モデルを設定し，ページ入出力実行時間を評価関数として，これを最小化する最適解について論じた．提案した集中方式は，実用的条件下で最適解を与える．また，より弱い条件のもとで従来の浮動方式より優れ，さらに条件によらず常に従来の固定方式より優れた入出力効率が期待できる．

本方式を実験的に実現し，浮動方式と性能を実測比較した．ページ入出力実行時間について，平均値で約38% の顕著な短縮が観察された．

本方式はすでに当社の大型オペレーティング・システム VOS 3 において実用化されている．今後は，本方式の効果とシステム全体の処理能力，応答性との関連[8]につき，考察を加える予定である．また，本論文では二次記憶のスペース利用率とページ入出力効率の関係は扱わなかった．スペース利用率が上がるにしたがってスロット割り当ての自由度が減るので，どの方式も固定方式に近くなり効率差異は減少すると予想されるが，今後機会があれば検討を加える予定である．

本論文は TSS アプリケーションを想定したが，バッチ・アプリケーションに対しても本論文の議論は成立する．ただしこの場合スワッピングの頻度が低いので，性能上の差異は比較的小さいと考えられる．また，今後主記憶と二次記憶の間に半導体，バブルなどによる中間記憶が介在してくる可能性がある．この場合でも中間記憶と二次記憶との間の情報転送に着目すれば本論文の議論は成立する．ただし参照頻度の高いページは中間記憶に保持されるので，二次記憶のスロット割り当て方式がシステム性能に与える影響は減少すると考えられる．

終りに，本研究についてご指導いただいた東京大学大須賀節雄助教授，ならびに本研究の機会を与えて下さった当社システム開発研究所三浦武雄所長，同ソフトウェア工場服部陽一部長，の諸氏に深く感謝いたします．

参考文献

1) 西垣 通，緒方慎八：仮想メモリ・システムの二次記憶管理方式の比較解析，情報処理学会論文誌，Vol, 20, No. 4, pp. 290-298 (1979).
2) Denning, P. J.: Effects of scheduling on file memory operations, Proc. SJCC, pp. 9-21 (1967).
3) Teory, T. J. and Pinkerton, T. B.: A Comparative Analysis of Disk Scheduling Policies, Comm. ACM, Vol. 15, No. 3, pp. 177-184 (1972).
4) Belady, L. A.: A study of replacement algorithms for a virtual-storage computer, IBM Systems Journal, Vol. 5, No. 2, pp. 78-101 (1966).
5) Denning, P. J.: The Working Set Model for Program Behavior, Comm. ACM, Vol. 11, No. 5, pp. 323-333 (1968).
6) Denning, P. J. and Graham, G. S.: Multiprogrammed Memory Management, Proc. IEEE, Vol. 63, No. 6, pp. 924-939 (1975).
7) Masuda, T.: Analysis of Memory Management Strategies for Multiprogrammed Virtual Storage Systems, Journal of Information Processing, Vol. 1, No. 1, pp. 14-24 (1978).
8) 西垣 通：多重プログラミング・システムにおけるフィードバック概念にもとづく一般資源管理方式，情報処理，Vol. 19, No. 11, pp. 1026-1033 (1978).
9) W. フェラー：確率論とその応用(上)，河田龍夫監訳，紀伊国屋書店 (1969).
10) 日立製作所：H-8549-2 ディスク制御装置 H-8589 ディスク駆動装置，ハードウェア・マニュアル，8080-2-007 (1974).
11) 日立製作所：VOS 2 システムプログラマの手引，プログラム・マニュアル，8080-3-002 (1975).
12) 日立製作所：VOS 3 システムプログラマの手引，プログラム・マニュアル，8090-3-002 (1976).
13) IBM: Introduction to Virtual Storage in System/370, Student Text (1972).

付 録

x と y の関係

x 個のブロックを次々と蓄積するとき，各試行でディスク・ヘッドの位置は $1 \sim L$ の一様分布にしたがう．y は，x 回の試行で少なくとも1回ディスク・ヘッドが停止していた相異なるシリンダ数の期待値である．ここで，全くディスク・ヘッドが停止しなかった相異なるシリンダの数が m である確率 $P_m(x, L)$ は次式で与えられる．証明は 9) の IV章 pp. 127-137 を参照されたい．

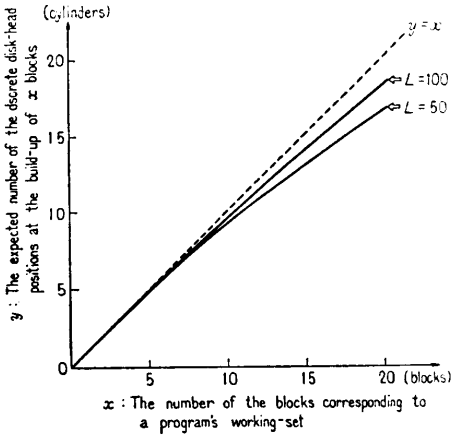

図 10 確率モデルにより求めた x と y の関係
Fig. 10 x versus y, calculated by the probabilistic model.

$$P_m(x, L)=\binom{L}{m}\sum_{i=0}^{L-m}(-1)^i\binom{L-m}{i}\left(1-\frac{m+i}{L}\right)^x$$
$$(\max(0, L-x)\leq m\leq L-1) \quad (22)$$

このときyは次のように求められる.

$$y=\sum_{m=\max(0, L-x)}^{L-1}(L-m)P_m(x, L) \quad (23)$$

実際の計算には，直接 (22) を用いず，x, mを変数とする次の漸化式を用いる方が簡便である[9].

$$\begin{cases}P_m(x+1, L)=P_m(x, L)(L-m)/L \\ \qquad\qquad +P_{m+1}(x, L)\cdot(m+1)/L. \quad (24)\\ (\max(0, L-x)\leq m\leq L-2)\end{cases}$$

$$\begin{cases}P_m(x+1, L)=P_{m+1}(x, L)\cdot(m+1)/L \\ (m=L-x-1)\end{cases} \quad (25)$$

式(25)はディスク・ヘッド位置が全く重複しない場合にあたる．また $m=L-1$ の場合は直接(22)より，

$$P_{L-1}(x, L)=L^{1-x} \quad (26)$$

である．式(24)(25)(26)より，x, m を1から順次増して $P_m(x, L)$ を求め，さらに式(23)よりyを求められる．$L=50$，100におけるxとyの関係を図10に示す．

(昭和54年3月28日受付)
(昭和55年5月15日採録)