

資料

ディスク装置へのファイルの適正配置法*

近藤 秀文** 吉田 郁三** 加藤 孝***

Abstract

An method, which gives the effective file allocation of multiple files onto multiple disc packs, is introduced in this paper.

This method decides the effective file allocation as reducing the total seek time using five characteristics: 1) Access frequency of each file, 2) Access transition between files, 3) Distribution unbalance of access cylinder addressing in each file, 4) Cylinders and their frequencies of movement of magnetic head in each file, and 5) Volume of each file, in a file application system.

There are three kinds of access form about files on disc packs: a) Serial access, b) Parallel access, and c) Mixed access of above two access forms. This paper treats of the case of the serial access form. An heuristic algorithm is used as a file allocation technique.

The reduction of total file access time from the result of calculation by this algorithm was coincident in enough accuracy with that of experiment. Then it is considered that this method is sufficiently available for practical use.

1. ま え が き

計算機システムの処理能力、たとえば時間当りの処理件数などは、その構成要素であるハードウェア、およびソフトウェアの固有の性能に依存するほかに、これら固有の性能を充分発揮させる利用技術（ファシリティ・マネジメント技術）によって左右されることはよく知られている。

本報告は、ファシリティ・マネジメント技術の1つの試みとして、データ・ベース等の格納媒体であるディスク装置に着目し、業務特性に応じて、情報の検索・格納時間を減少させるように、ファイルをディスク装置上に適正に配置するためのファイル配置法に関するものである。

1つのジョブで使用するディスク装置は普通、複数のディスク・パックから構成されている。このディスク・パック上へのファイルの配置法について、従来、1つのディスク・パック上への複数ファイルの配置法¹⁾については、研究がなされているが****、複数ディスク・パック上への複数ファイルの配置法については報告がみられない。

本報告は、次に述べるようなアプローチで、複数ディスク・パック上への複数ファイルの配置法を与えたものである。すなわち、データ・ベース・システム等のように大容量ディスクに大量の情報を格納し、検索・更新を行うシステムにおいては、

- (1) 各ファイルに対するアクセス頻度
- (2) ファイル間のアクセス推移
- (3) 各ファイル内のアクセス・シリンダ・アドレスの偏り
- (4) 各ファイル内での磁気ヘッドの移動シリンダ数とその頻度
- (5) 各ファイルの容量

に特徴（偏り）があるとき、この特徴を総合的に利用

* An Effective File Allocation Method onto Disc Devices, by Hidefumi KONDO, Ikuzo YOSHIDA (System Development Laboratory, Hitachi Ltd.) and Takashi KATO (Software Works, 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

**** 単一ディスク・パック上へのファイル配置については、Bool & Babbage 社(米)および COMTEN 社(米)が配置プログラムを開発している。

することによって、ファイル・アクセス時の磁気ヘッドの総移動量すなわち、トータル・シーク・タイムを減少させることができる。このような考えに基づいて開発したヒュリスティックなアルゴリズムによるファイルの適正配置法について述べる。

計算機システムにおいて、ディスク・パックに格納されたファイルへのアクセス形態は、

- (1) シリアル・アクセス形態：複数のディスク・パック中のファイルに対して時系列的にシリアルにアクセスを行う形態
- (2) パラレル・アクセス形態：複数個のディスク・パック中のファイルに対して同時にパラレルにアクセスする形態
- (3) 両者の混合形態

の3種類に分類できる。本報告は、上記の(1)について、評価尺度としてシーク・タイム(磁気ヘッドの位置決め時間)を設定し、このシーク・タイムを減少させるファイルの適正配置法について述べたものである。この方法を逐次統合法と呼ぶ。

本報告では、ファイルは、容量(シリンダ数)をもつ1つの個体と考え、同一ファイルは同一ディスク・パック上にもみ格納でき、かつそのディスク・パック上で連続した領域に格納されるものとする。

2. 逐次統合法

逐次統合法は、Fig. 1に示すように、(a)シーク・タイムの計算を行う部分と、(b)適正配置の手順の部分とから成る。この図に示すように、逐次統合法は、

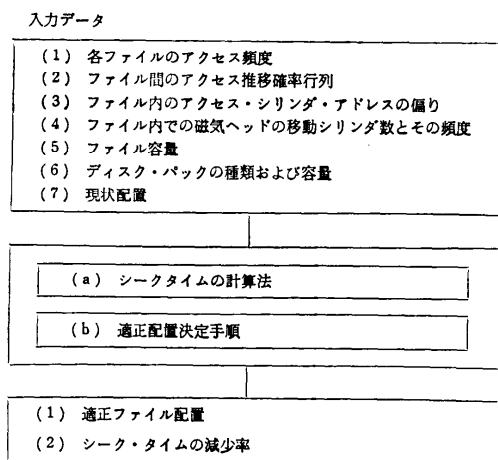


Fig. 1 Effective file allocation method using automatic synthesis

7種類の入力データを与えると結果として適正なファイル配置とシーク・タイムの節減効果を出力する。以下にシーク・タイムの計算法と適正配置の手順について述べる。

2.1 シーク・タイムの計算法

ここでは、評価尺度であるトータル・シーク・タイムが任意の配置状態において計算可能であることを示す。

2.1.1 論理的アクセス推移頻度とファイル間距離

ファイルに対するアクセスは、時間的経過、 $t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4 \rightarrow t_5$ に従って、たとえば Fig. 2のようにアクセスされるであろう。

これを各ファイル間の論理的アクセス推移と考えると、Fig. 3(a)のような論理的アクセス推移確率行列 P を作成する。 p_{ij} の意味は、ファイル F_i をアクセスし、次にファイル F_j をアクセスする確率が p_{ij} であることを示す。

また、各ファイルに対するアクセス頻度を Fig. 3(b)のように論理的アクセス頻度行列 N として表わす。 P と N との積は、Fig. 4(a)のような論理的アクセス推移頻度行列 Q となる。 q_{ij} の意味は、ファイ

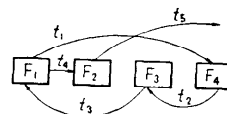


Fig. 2 Logical transition of historical file access

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{bmatrix} \quad N = \begin{bmatrix} N_1 & & & \\ & N_2 & & 0 \\ & & \ddots & \\ 0 & & & N_n \end{bmatrix}$$

(a) Logical access transition probability matrix

(b) Access frequency matrix

Fig. 3 Logical access transition probability between files and access frequency of each file

$$Q = N \cdot P = \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{bmatrix} \quad L = \begin{bmatrix} l_{11} & l_{12} & \dots & l_{1n} \\ l_{21} & l_{22} & \dots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}$$

(a) Logical access transition frequency matrix

(b) Distance matrix

Fig. 4 Logical transition frequency and distance between files

ル F_i の次にファイル F_j を論理的にアクセスする推移頻度が q_{ij} 回であることを示す。さらに、Fig. 4 (b) のようなファイル間距離行列 L を定義する。

L の要素 l_{ij} は、磁気ヘッドがファイル F_i から F_j に移動するときの平均的な移動距離を表わす。この設定方法については 2.1.3 で述べるが、ディスク・パック上へのファイルの物理的な配置が定まらなると決定できないものである。トータル・シーク・タイムは、基本的には、ファイル F_i, F_j 間の論理的アクセス推移頻度 q_{ij} とファイル間距離 l_{ij} を用いて計算する。しかし、複数ファイルを複数ディスク・パックに分散配置したときには、磁気ヘッドの動きを考慮するために、各ディスク・パックごとに物理的なアクセス推移頻度を計算することが必要になる。これについて、2.1.2 に述べる。

2.1.2 物理的アクセス推移頻度行列の作成

ディスク・パックごとのファイル間の物理的アクセス推移頻度は、2.1.1 で述べた論理的アクセス推移頻度および推移確率を基にして計算する。この2つはデータとして収集可能である。

ファイルが複数のディスク・パックに分散して、配置されたときには、ファイル間距離の定義できない場合が生ずる。たとえば、Fig. 5 のような配置の場合には、ファイル F_i と F_k 、 F_j と F_k などの距離は、定義できない。そこで、次のような方法で、ファイル間の論理的アクセス頻度行列 Q を、同一ディスク・パック上に配置されたファイルだけについて、ファイル間推移頻度が表記されるように変形する。

Fig. 5 において、最初磁気ヘッドが、図中のディスク・パック D_1 、および D_2 の▲印で示す位置にあったとする。ここで、 t_1, t_2 のような論理的アクセス推移が起きると、磁気ヘッドは、図中の点線で示すように、 $F_i \rightarrow F_k$ の論理的推移に対しては $F_i' \rightarrow F_k$ 、 $F_k \rightarrow F_j$ に対しては $F_i \rightarrow F_j$ のように物理的な移動を起こす。この移動は、論理的アクセス推移とは全く別なものであり、複数ファイルを複数ディスク・パックに分散配置したために $F_k' \rightarrow F_k$ および $F_i \rightarrow F_j$ に関して実質的に増加した磁気ヘッドの物理的推移である。こ

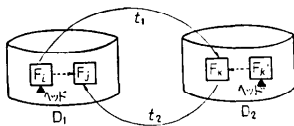


Fig. 5 Logical access transition and physical access transition

の物理的推移頻度を q_{ij}' で表わす。 q_{ij}' の算定にあたっては、次の仮定を行う。いったん、あるファイル F_k をアクセスした後、次にファイル F_j をアクセスする確率は、論理的アクセス推移確率 p_{kj} によってのみ決まるものとする。すなわち、いま、どのファイルにアクセスしているかを「状態」としたマルコフ連鎖を仮定する。以上から、増加の物理的推移回数 q_{ij}' は、 $F_i \rightarrow F_k$ の論理的推移頻度 q_{ik} と論理的推移確率 p_{kj} との積 $q_{ik} p_{kj}$ として計算することができる。増加の物理的推移 $F_i \rightarrow F_j$ は、論理的推移が $F_i \rightarrow F_k \rightarrow F_j$ だけでなく、 $F_i \rightarrow F_k \rightarrow F_k' \rightarrow F_j, \dots$ などによっても起こりうるので、結局、増加の物理的推移頻度 q_{ij}' は、次のように計算することができる。

$$q_{ij}' = \sum_{i,j \in x} q_{ik} p_{kj} + \sum_{k \in x} \sum_{k' \in x} q_{ik} p_{kk'} p_{kj'} + \dots \quad (2.1)$$

X : ある1つのディスク・パック上に配置されているファイルの集合

また、Fig. 5 で論理的推移 $F_i \rightarrow F_j$ が起これば、勿論、物理的推移 $F_i \rightarrow F_j$ が起こるので、全体としての物理的推移頻度は、次式で計算できる。

$$q_{ij}^{(+)} = q_{ij} + q_{ij}' \quad (2.2)$$

以上述べたように、互いに別々のディスク・パック上に配置されているファイル間の論理的推移は、同一ディスク・パック上のファイル間の物理的推移に置き換えられることができたのである。また、このような置き換えによって、互いに別々のディスク・パックに配置されているファイル F_i, F_k 間の物理的推移頻度 $q_{ik}^{(+)}$ は、

$$q_{ik}^{(+)} = 0 \quad (2.3)$$

である。例を Fig. 6 に示す。 $Q_1^{(+)}$ 、 $Q_2^{(+)}$ 、および $Q_3^{(+)}$ は、それぞれ、3つのディスク・パック上に配置された各ディスク・パック上でのファイル間の物理的推移頻度を表わす。

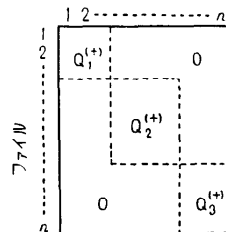


Fig. 6 Physical access transition frequency matrix $Q^{(+)}$

2.1.3 ファイル間距離の設定

磁気ヘッドの移動を伴う物理的推移頻度 $q_{ij}^{(*)}$ は、同一ディスク・パック上に配置されたファイル間のみ値をもち、他の場合には、 $q_{ij}^{(*)}=0$ であるから、ファイル間の距離も同一ディスク・パック上に配置されたファイル間のみについて設定すれば充分である。

ファイル間距離は、ファイル F_i の容量 l_i (シリンダ), 論理的アクセス推移に伴う、ファイル内の相対的シリンダアドレス t のアクセス頻度 f_{it} , および論理的アクセス推移に伴うファイル内移動について、その移動量が k シリンダである頻度 n_k^i を基にして計算する。この3つは、データとして収集可能である。以下詳しく説明する。

ファイル間距離の設定は、次の3つの場合に分けて計算する。

- ・ 隣接する2つのファイル F_i と F_j 間の平均移動距離
- ・ 隣接しない2つのファイル F_i と F_j 間の平均移動距離
- ・ 1つのファイル F_i 内での平均移動距離

(1) 2つのファイル間の平均移動距離

実際のシステムから、2つのファイル F_i および F_j に対するシリンダのアクセス状態が Fig. 7 のグラフのようにデータ収集できたとする。そこで、磁気ヘッドが、平均的に F_i のどの位置から F_j に向って移動するかを次の計算によって算出する。すなわち、この平均的な位置はアクセス頻度に依存するものとする。

$$l_{ix} = l_i - \sum_{t=1}^{l_i} \rho_{it} \cdot t$$

$$\text{ただし } \rho_{it} = f_{it} / \sum_{s=1}^{l_i} f_{is} \quad (2.4)$$

また、Fig. 7 の中の l_{xj} は、 F_j にアクセスが行われたとき、平均的に F_j のどの位置にアクセスが起きるかを表わすものであり、 l_{ix} と同様に次の計算によって算出する。

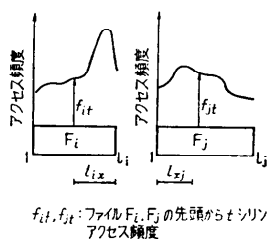


Fig. 7 Access frequency of cylinder address in each file

$$l_{xj} = \sum_{t=1}^{l_j} \rho_{jt} \cdot t$$

$$\text{ただし } \rho_{jt} = f_{jt} / \sum_{s=1}^{l_j} f_{js} \quad (2.5)$$

2つのファイル F_i と F_j が隣接する場合には、上記の l_{ix} と l_{xj} を用いて、 F_i から F_j への平均移動距離 l_{ij} は、次のように計算する。

$$l_{ij} = l_{ix} + l_{xj} \quad (2.6)$$

逆に F_j から F_i への平均移動距離は、 $l_{jx} = l_{xj}$, $l_{xi} = l_{ix}$ とし、 $l_{ij} = l_{jx} + l_{xi}$ によって計算する。すべてのファイルについて、 $l_{ix} (i=1, \dots, n)$ を求めておけば、任意の2つのファイル間の平均移動距離の計算が可能となる。

また、ファイル F_i と F_j が隣接せずに、中間に複数のファイル F_k (容量 l_k , ($k=1, 2, \dots$)) が配置されている場合には、 F_i から F_j への平均移動距離を、次式によって計算する。

$$l_{ij} = l_{ix} + \sum_k l_k + l_{xj} \quad (2.7)$$

(2) 1つのファイル内での平均移動距離

実際のシステムから、ファイル・アクセス・データを収集するとき、同一ファイル内の移動に関しては、Table 1 のような形式で、移動距離毎に頻度 n_k^i を収集する。そして、次の計算法によって、平均移動距離ではなく平均シーク・タイムを求める。

$$t_{ii} = \sum_{k=0}^{l_i-1} p_k^i f(k)$$

$$\text{ただし } p_k^i = n_k^i / \sum_{k=0}^{l_i-1} n_k^i \quad (2.8)$$

上式の中で $f(k)$ は、移動距離からシーク・タイムへの変換関数であり、Fig. 8 に示すようなものである。

Table 1 Cylinders and their frequency of movement of magnetic head in file. Fi

移動距離	0	1	2		l_i-1
頻度	n_0^i	n_1^i	n_2^i		$n_{l_i-1}^i$

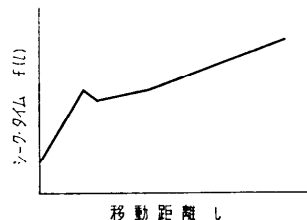


Fig. 8 Seek time function

る。後で述べるトータル・シーク・タイムの計算で、ファイル内移動についても、扱いかを統一するために、便宜上(2.9)式で示すように平均シーク・タイム t_{ii} を平均移動距離 l_{ii} に変換しておく。

$$l_{ii} = f^{-1}(t_{ii}) \quad (2.9)$$

2.1.4 トータル・シーク・タイムの計算

トータル・シーク・タイムの計算は、2.1.2, 2.1.3で述べた $q_{ij}^{(+)}$, l_{ij} , l_{ii} , およびシーク・タイム変換関数を用いて、次のような計算式で求める。

$$T = \sum_{i,j} q_{ij}^{(+)} f(l_{ij}) \quad (2.10)$$

以上の説明で述べたように、あらかじめ、データとして、 P , Q , f_{ii} , および n'_k を得ていれば、 $q_{ij}^{(+)}$, l_{ij} , および l_{ii} を求めることができ任意のファイル配置状態におけるシーク・タイムを(2.10)式に従って計算することができる。次に、この計算法を使用した適正配置の手順について述べる。

2.2 適正配置の手順

任意のファイル配置状態におけるトータル・シーク・タイムの計算方法が可能であることを示したので、論理的には、全ての配置の組み合わせについてトータル・シーク・タイムを求めることによって、最適配置を決定できる。配置の組み合わせは、次の2段階で発生する。

(a) ファイルのディスク・パックへの振分け

(b) ディスク・パック内でのファイルの並べ方

しかし、これらの全ての組み合わせは膨大な数になるので、概略次のような手順によって、トータル・シーク・タイムがより少ない配置に、逐次近づけていく方法を考案した。

(i) 任意の状態配置におけるトータル・シーク・タイムを T とする。

(ii) この状態から後述する方法で、あるファイルを他のディスク・パックへ移動するか、または他のディスク・パック内のファイルと変換する。

(iii) 変動のあったディスク・パック内のファイルについてファイルの並べ方を変える。

(iv) 上記(ii)および(iii)の操作によって T の減少が得られるとき、その配置を選ぶ。

(v) 減少効果がある限り、(ii), (iii), (iv)の操作を繰り返す。

このように、ある配置から少なくともそれより良い配置へと逐次良い方向へ進める方法である。

次にこの手順の詳細について述べる。まず、ディス

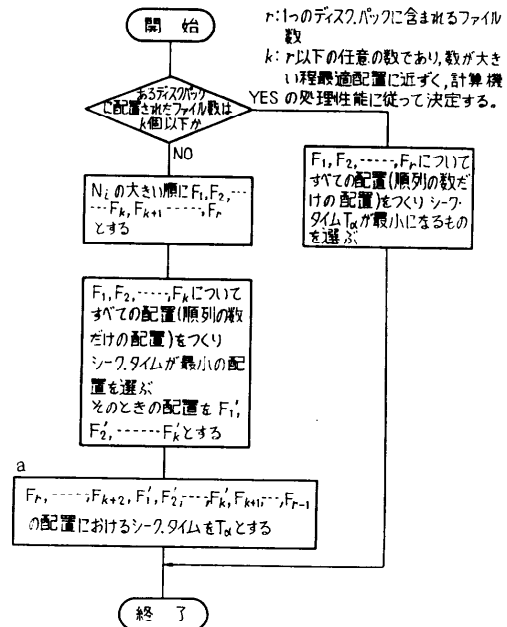


Fig. 9 Procedure, deciding the effective file allocation in each disc pack

ク・パック D_a ごとのシーク・タイム T_a について定義する。1つのディスク・パック D_a に複数個のファイルが配置されているときには、そのディスク・パックに関して、シーク・タイムが最小になる配置のシーク・タイムを T_a とする。具体的には、Fig. 9 のフローチャートに従って T_a を求める。このとき、 T_a は、 D_a 内のファイルを対象として(2.10)式を用いて計算する。図中、aの部分において k 個を越えるファイルについては、順次、両側に配置したが、これは文献1)の単一ディスク・パック上への最適配置法によるものである。文献1)の方法は、各ファイルの容量が等しく、かつアクセス推移確率がアクセス頻度に従うという条件が必要であるが、ここでは、ファイル数が k 個を越えたものについては、近似的に文献1)の方法を採用した。これは、1つのディスク・パック上のファイル数が大きくなると、ファイルの並べ方の数が膨大になるため、このような方法をとったものであり、計算機の性能に従って k の値を設定する。

適正配置の手順は、以下の4つのステップで行う。

(1) 初期設定

n 個の全てのファイルを m 個のディスク・パックの外に出しておき、このとき磁気ヘッドの移動に要するトータル・シーク・タイム T が最大であるとする。そ

して、 τ_i をファイル F_i のトータル・シーク・タイムとして、 T は次のように設定する。

$$\begin{aligned} T &= \sum_{i=1}^n \tau_i \\ &= \sum_{i=1}^n N_i t \end{aligned} \quad (2.11)$$

t は、ファイルをディスク・パックの外に出したために、各ファイルに対して 1 回アクセスが行われるときの回りのシーク・タイムである。 t は、かなり大きな値、たとえば 250 ms のように設定する。

(2) 格 納

ディスク・パックの外に出したファイルの中から任意にファイル F_i をとり、任意のディスク・パック D_α に配置したときの T の減少分 $\Delta T_{i\alpha}$ の計算を試みる。減少分 $\Delta T_{i\alpha}$ が最大となるファイル F_i をディスク・パック D_α に格納する。ここで、 $\Delta T_{i\alpha}$ は以下の計算によって求める。

$$\Delta T_{i\alpha} = (\tau_i + T_\alpha') - T_\alpha \quad (2.12)$$

ただし、 T_α' は F_i を D_α に入れる前の D_α に関するトータル・シーク・タイムであり、 T_α は F_i を D_α に入れた後の D_α に関するトータル・シーク・タイムである。格納するファイルが決まったとき、 T をあらためて下記のようにする。

$$T = T - \Delta T_{i\alpha} \quad (2.13)$$

この操作を繰り返してゆくと、 n 個のファイルがすべて m 個のディスク・パック中に配置される。このときのトータル・シーク・タイムは次のようになる。

$$T = \sum_{\alpha=1}^m T_\alpha \quad (2.14)$$

(3) 移 動

ディスク・パック D_α 上に配置されたファイルの中の任意なファイル F_i を、別なディスク・パック D_β に移動し、 T の減少分 $\Delta T_{i\beta}$ を計算する。ただし、移動操作でディスク・パックをオーバー・フローする場合は計算を除外する。 $\Delta T_{i\beta} > 0$ であるものの中で最大の $\Delta T_{i\beta}$ を選び、そのファイル F_i をディスク・パック D_β に移動する。すべての $\Delta T_{i\beta}$ が、 $\Delta T_{i\beta} \leq 0$ ならば、このステップを終了する。 $\Delta T_{i\beta}$ は以下のように計算される。

$$\Delta T_{i\beta} = (T_\alpha' + T_\beta') - (T_\alpha + T_\beta) \quad (2.15)$$

ここで、 T_α' 、および T_α は、それぞれ D_α から F_i を移動前、および移動後についての D_α に関するトータル・シーク・タイムであり、 T_β' 、および T_β は、それぞれ F_i を D_β 上に移動前、および移動後の D_β

に関するトータル・シーク・タイムである。

(4) 交 換

ディスク・パック D_α 上の F_i と D_β 上の F_j の交換を試み、 T の減少分 $\Delta T_{ija\beta}$ を計算する。このとき、 F_i と F_j のファイル容量は異なってもよいが、交換によってどちらかのディスク・パックがオーバー・フローを起こす場合には計算を除外する。 $\Delta T_{ija\beta} > 0$ であるものの中で最大の $\Delta T_{ija\beta}$ を選びそのファイル F_i と F_j を交換する。

$\Delta T_{ija\beta}$ は以下のように計算される。

$$\Delta T_{ija\beta} = (T_\alpha' + T_\beta') - (T_\alpha + T_\beta) \quad (2.16)$$

ここで、 T_α' および T_α は、それぞれ、 D_α 上の F_i と D_β 上の F_j を交換前および交換後の D_α に関するシーク・タイムであり、 T_β' および T_β は、同様にそれぞれ交換前および交換後の D_β に関するシーク・タイムである。すべての $\Delta T_{ija\beta}$ が、 $\Delta T_{ija\beta} \leq 0$ ならば、このステップは終了であり、このとき、各ディスク・パックに格納されているファイル配置が本配置法による適正配置となる。

3. 実 験

本配置法の目的は、シーク・タイムを減少させることである。本方法を用いた場合、実際にどの程度シーク・タイムを減少できるものかを調べるため、某工場におけるオンライン・システムを対象として実験を行った。このシステムは、ディスク・パックが 6 台、ファイルが 38 個あったが、99% 以上のアクセスが 11 個のファイルに集中していたため、逐次統合法の適用は、この 11 個のファイルを対象とした。このシステムからファイルに対するアクセス・データを収集し、Table 2 および Table 3 (次頁参照) にまとめた。Table 2 は、各ファイルに対するアクセス頻度、Table 3 は、ファイル間の論理的アクセス推移確率行列である。Fig. 10 (次頁参照) はこのときのファイル配置状態を示したものである。Fig. 11 (次頁参照) は、これらのデータを基にして逐次統合法を適用し決定した適正配置である。Table 4 (次頁参照) は、現在の配置と適正配置とのトータル・シーク・タイムについて試

Table 2 Example of access frequency

ファイル名	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆
頻 度	5252	4609	1956	1956	7480	1508
ファイル名	F ₇	F ₈	F ₉	F ₁₀	F ₁₁	
頻 度	5119	5971	4846	3221	2362	

	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆		
F ₁		.040	.169	.106	.025	.102	.025	F ₆
F ₂								F ₇
F ₃								F ₈
F ₄								F ₉
F ₅								F ₁₀
F ₆								F ₁₁
	F ₆	F ₇	F ₈	F ₉	F ₁₀	F ₁₁		

Table 3 Example of logical access transition probability

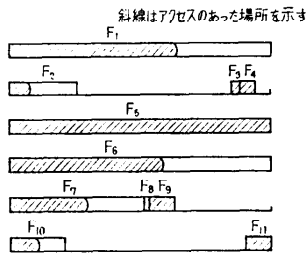


Fig. 10 Allocation before applying the effective file allocation method

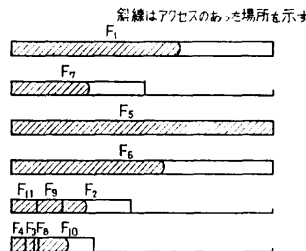


Fig. 11 Effective file allocation

算した結果を比較したものである。Table 4 からわかるように、この実験ではトータル・シーク・タイムの約 33% を減少できることがわかった。逐次統合法は、特定の期間（たとえば、1 日分）のファイル・アクセス・データに基づいて、そのデータから得たファイル・アクセスの特徴に合わせて適正配置を行うものである。したがって、ファイル・アクセスの特徴が長期間持続されないと効果がうすれてくる。Table 5 はオンライン・システムにおける a, b, c, d, e, f の任意の 6 日についてデータを収集し、 a 日のファイル・アクセス・データに基づいて適正配置を決定し、その配置状態のままで、 $a \sim f$ の各日について効果を調べたものである。Table 5 では、 a 日の節減率を 100 とし

Table 4 Comparison of total seek time

	トータル・シーク タイム (sec)	比率	節減率
従来配置	2273.386	100	—
適正配置	1497.222	67.2	32.8%

Table 5 Reduction ratio of total seek time

数値は a 日を 100 としたときの相対値

日付	a	b	c	d	e	f
シーク・タイム 節減率	100	122	97	127	116	120

$a \sim f$ の各日について、節減効果を比較したものである。Table 5 からわかるように、この実験対象となったオンライン・システムの場合には、節減効果が安定していることがわかる。本方法を適用する場合には、各システムに対して同様な検査が必要である。

4. シーク・タイムの計算精度

本配置法では、シーク・タイムは、すべて計算によって求めている。そこでこの計算精度の検証を以下の手順で行った。

- (1) ディスク装置上に実システムと同様なファイルのモデルを作成する。
- (2) 実システムから収集したファイル・アクセス・データに従って、実際に全く同じ順序で、モデル・ファイルをアクセスし、ハードウェア・モニタによって、シーク・タイムを測定する。測定は、7 種類のファイル配置について行った。

Table 6 は、この実験による計算値と実測値を比較したものである。この結果から、計算値の誤差は 10% 未満であることがわかる。

Table 7 (次頁参照) は、4. で述べた実験システムの従来配置と適正配置の両者について、同様な実験を行った結果である。この Table 7 から、計算値および実測値のいずれの結果から、適正配置に変えることにより 33%~34% のトータル・シーク・タイム節

Table 6 Accuracy of seek time calculation

ケース	計算値 (sec)	実測値 (sec)	誤差 (%)
1	519.128	512.137	1.4
2	738.269	755.014	-2.2
3	337.726	311.395	8.5
4	329.663	334.935	-1.6
5	190.227	184.574	3.1
6	367.514	376.744	2.4
7	314.474	304.841	3.2

Table 7 Comparison between calculation and measurement about total seek time

	計 算	実 測
a. 従来配置のトータル・シーク・タイム (sec)	2273.386	2257.038
b. 適正配置のトータル・シーク・タイム (sec)	1497.222	1497.411
c. 比率 $\frac{b}{a} \times 100$ (%)	67.2	66.3
d. 節 減 率 $1-c$ (%)	32.8	33.7

減効果があることがわかる。

5. む す び

ディスク装置上に格納されたファイルに対してシリアルにアクセスが行われる場合について、そのアクセス状態を利用して磁気ヘッドのトータル・シーク・タイムを減少させるファイルの適正配置法として逐次統合法を開発した。逐次統合法の効果を検証するため、実システムからファイル・アクセス・データを収集し、適用試算の実験を行った結果、そのシステムの場合

合にはトータル・シーク・タイムを約30%減少できることがわかった。また本配置法の計算精度を検証するため、実システムから収集したファイル・アクセス・データに基づいて精度検証のためのモデル・ファイルにアクセスさせ、ハードウェア・モニタによってシーク・タイムを測定した。この実験から計算値の計算誤差は10%未満であることがわかった。以上の結果から、この方法は、実用的なファイル配置法としての活用が可能であると考えられる。

終りに本研究を御指導下さった東京工業大学の榎本肇教授に深謝致します。

参 考 文 献

- 1) D.D. Grossman 他: Placement of Records on a Secondary Storage Device to Minimize Access Time, Journal of ACM. Vol. 20, No. 3, pp. 429~438 (1973).

(昭和51年3月26日受付)

(昭和51年11月15日再受付)