入出力時間から見たディスクキャッシュの効果について

小畑征二郎，松沢茂，宮崎正俊（東北大学）
表　俊夫，神山典，高橋壮幸（日本電気）

１．はじめに

ディスクキャッシュ・システムはディスク装置と主記憶装置間に高速なバッファメモリを設け，ディスク装置に対するアクセス時間をみかけ上短縮するためのものである。このバッファメモリをディスクキャッシュ・メモリ（以下，ディスクキャッシュという）といい，ディスク制御装置側におくのが一般的であるが，ＡＣＯＳ－６（日本電気のＡＣＯＳシステム１０００系のオペレーティング・システム）のディスクキャッシュ・システムでは，中央処理装置側においている。本稿では，ＡＣＯＳ－６のディスクキャッシュ・システムの概要とディスク装置に対する入出力時間から見たディスクキャッシュの効果のついて，東北大学大型計算機センターに設置されているＡＣＯＳ－１０００の計測結果を基に報告する。

２．ＡＣＯＳ－６のディスクキャッシュ・システムの概要

ＡＣＯＳ－６のディスクキャッシュ・システムでは，図１に示すようにディスクキャッシュを中央処理装置側に置き，その制御を入出力処理装置（以下，ＩＯＰという）上のソフトウェアであるディスクキャッシュ・ハンドラ（以下，ＤＣＨという）が行っている。つまり，ＤＣＨはＡＣＯＳ－６と連携して，ディスクキャッシュの使用，不使用の判断，目的のデータがディスクキャッシュ上に有る場合（以下ヒットという）とない場合（以下，ヒットミスという）の制御，ディスク制御装置に対する入出力の指示およびディスクキャッシュの利用に関する統計情報の収集などを行っている。

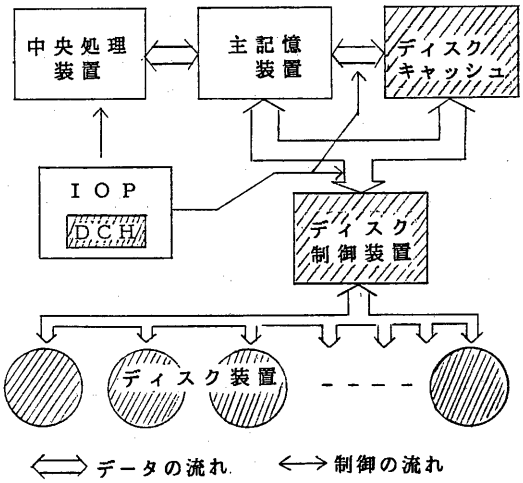

図１　ディスクキャッシュ・システムの概念図

ディスクキャッシュは，容量を最大３２ＭＢまでもつことができ，図２に示すようにブロックに分割されている。このブロックは，ディスクキャッシュを管理するときの最小単位で，ディスクキャッシュとディスク装置間のデータの転送単位でもある。ブロックはシステムの環境に応じて，４ＫＢ，８ＫＢ，１６ＫＢの大きさに設定できる。このブロックの中には複数個の論理レ

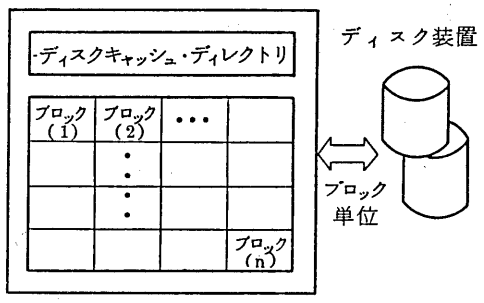

図２　ディスクキャッシュの構造

コードが含まれる。また，ディスクキャッシュ・ディレクトリはブロックを管理するためのもので，これには各ブロックに含まれるレコードのディスク上のアドレスに関する情報が記録されている。ＤＣＨは，この情報を参照することによって入出力レコードがディスクキャッシュ上にあるかどうかを判断する。またブロックの置換アルゴリズムはＬＲＵ（Ｌｅａｓｔ　Ｒｅｃｅｎｔｌｙ　Ｕｓｅｄ）方式で，一番昔に参照されたブロックの内容を新しいものと置き換える。

ディスクキャッシュを使用する場合には，ファイルの種別やアクセス特性に応じて，次の４種の動作モードを選択することができる。

（１）基本モード
（２）シーケンシャル・ファイル・モード
（３）一時ファイル・モード
（４）高速ファイル・モード

（１）の基本モードとはランダム・ファイルに適用されるもので，入力の場合には目的のレコードがディスクキャッシュ上にあれば，そのレコードをディスクキャッシュより読み出し，無い場合には，先ずディスクから目的のレコードを含むブロックを読み出してディスクキャッシュに書き込み，次にそのレコードをディスクキャッシュから読み出す。出力の場合には，レコードを書き込むブロックがディスクキャッシュ上にあれば，そのレコードをディスクキャッシュとディスクの両方に書き込み，ディスクキャッシュ上に無ければディスクにのみ書き込む。つまり，出力の場合には常にデータはディスクに書き込まれるのでアクセス時間は短縮されない。

（２）のシーケンシャル・ファイル・モードとは，シーケンシャル・ファイルのアクセス特性を考慮して，基本モードに次のような処理を追加したものである。入力の場合，ヒットミスすると実際に要求されたレコード長の何倍かのデータをディスクから読み出し，ディスクキャッシュに書き込んでおく。また，ブロックの最後のレコードが読み出されると，そのブロックを直ちに解放する。

（３）の一時ファイル・モードとは，一時ファイルでは書き込まれたデータは直ちに読み出されるという性質をもっているので，出力時にヒットミスをした場合でもデータをディスクキャッシュに書き込むようにしたものである。

（４）の高速ファイル・モードとは，必ずしもディスクに保存する必要のない一時ファイルに適用され，ディスクキャッシュと主記憶装置間だけでデータの入出力を行うものである。

３．計測システムと計測方法

東北大学大型計算機センターのシステム構成は，２ＣＰＵ（ＥＰＵ４台），主記憶装置５６ＭＢ，ディスク装置容量１８．５ＧＢ，ディスクキャッシュ８ＭＢなどからなり，通常の運用ではディスクキャッシュをコンパイラの入出力ファイルとワークファイル，システム・ライブラリ・ファイルおよびすべての用途のシーケンシャル・ファイルに適用している。また，ディスク装置とディスクキャッシュ間の転送単位（ブロック・サイズ）は１６ＫＢ（４ＫＷ）であり，その転送速度は２種類のディスク装置があるため，０．８ＭＢ／秒と１．２ＭＢ／秒である。

ところで，不特定多数のジョブを多重処理している計算センターでは，通常の運用時に正確なディスクキャッシュの効果を計測することは困難である。したがって，我々は計測

用のシステムを特別に作成し，運用時間帯外に計測のために作成したジョブを流して計測した。その計測用システムの構成概略を示すと図３のようになる。ディスク装置としては転送速度０．８ＭＢ／秒，容量２００ＭＢのもの８台を使用し，システム・プログラム・ファイル，システム・インプット・ファイルおよびシステム・アウトプット・ファイルを全装置に平均に割り当て，残りの部分すべてを利用者ファイル領域（利用者のプログラムやデータの保存，プログラム実行時のワークなどに使える領域）とした利用者のファイル領域は，全体で約1,２９８ＭＢである。

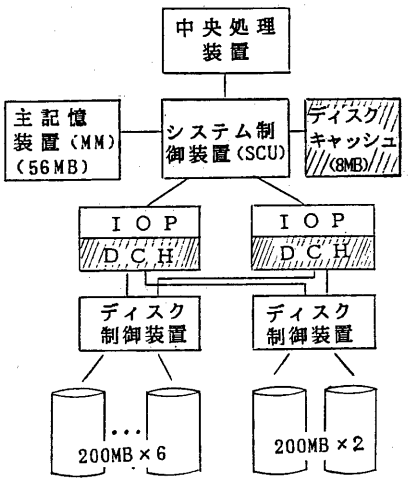

図３　計測システムの構成概略

計測用ジョブは，シーケンシャル・ファイルの入出力時間を計測するものとランダム・ファイルの入出力時間を計測するものをＦＯＲＴＲＡＮで作成した。入出力時間を測定する部分の流れを示すと図４のようになる。１回の入出力時間は，ＤＯの制御も含めた時間（Ｔ２－Ｔ１）をＮで割って求めている。また，ランダム・ファイルの大きさを３００ＬＬＩＮＫ（３８４ＫＢ）とし，入出力のレコードのアドレスは乱数を使って決めている。計測のためのシステムのタイプは，表１の○印の部分の８タイプとした。ディスクキャッシュの大きさを１２８ＫＢ，２５６ＫＢ，５１２ＫＢとしたのは，ランダム・ファイルの大きさが３８４ＫＢだからである。ディスクキャッシュの大きさが８ＭＢでブロック・サイズが１６ＫＢのものは，通常の運用時の形態である。また，ブロック・サイズの欄がＯＦＦのものはディスクキャッシュを使用しないものである。なお，計測用ジョブは各タイプともシングル・ストリームで実行して統計情報を収集した。

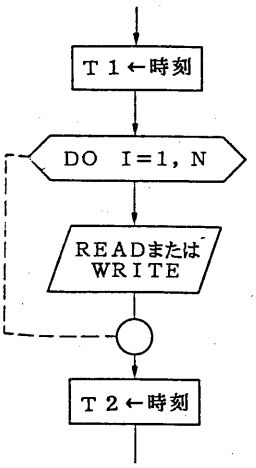

図４　入出力時間の測定の流れ図

表１　計測のためのシステム・タイプ

キャッシュサイズ ＼ ブロックサイズ	4KB	8KB	16KB	OFF
128KB	○	＊	＊	×
256KB	○	○	＊	×
512KB	○	○	○	×
8 MB	×	×	○	○

＊：システム的に構成不可

4．計測結果とその考察

4．1　論理レコード・サイズと入出力時間

論理レコード・サイズとはプログラムが直接出す入出力レコードのサイズである。図5と図6は，シーケンシャル・ファイルとランダム・ファイルの1入力当りの平均入出力時間とレコード・サイズの関係を出力の場合，ディスクキャッシュを使わない場合，およびディスクキャッシュ・サイズを128KB，256KB，512KB，8MBにした場合に分けて示したものである。ただし，ブロック・サイズは，128KB，256KB，512KBでは4KBとし，8MBでは16KBとしている。

これらの図より，次のことがわかる。

（1）入出力時間は，レコード・サイズが320W（＝1280B）大きくなるごとに段階的に増加する。これは主記憶装置とディスク装置間の物理的な転送が320W単位で行われるためである。

（2）出力の場合とディスクキャッシュを使わない入力の場合の入出力時間は，全く等しくなる。これは，出力の場合ヒット，ヒットミスにかかわらずディスクにデータを書きに行くためである。

（3）シーケンシャル・ファイルとランダム・ファイルの場合の比較を行うと，ランダム・ファイルの方がディスクキャッシュ・サイズが大きくなると，その効果も大きくなることがわかる。これは，シーケンシャル・ファイルとランダム・ファイルの制御の違いによるものである。一般に，シーケンシャル・ファイルではディスクキャッシュ・サイズよりもブロック・サイズによる影響の方が大きい。その例は後述する。

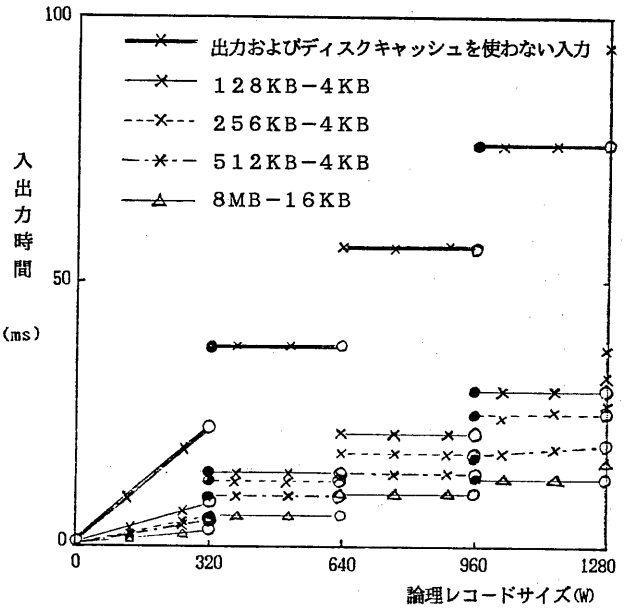

図5　シーケンシャル・ファイルにおけるレコード・サイズと入出力時間

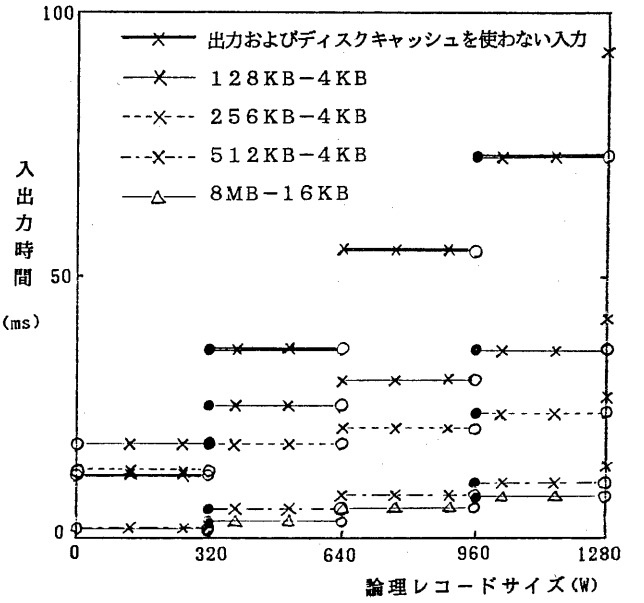

図6　ランダム・ファイルにおけるレコード・サイズと入出力時間

４．２　ディスクキャッシュ・サイズと入力時間

図７は，論理レコード・サイズが６４０Ｗと１２８０Ｗの場合のディスクキャッシュ・サイズと１入力当りの平均入力時間の関係を示したものである。図中の数字はブロック・サイズ（単位：ＫＢ）である。これによると，シーケンシャル・ファイルではディスクキャッシュ・サイズよりもブロック・サイズの方が入力時間に対する影響が大きく，ランダム・ファイルでは，ディスクキャッシュ・サイズの方が入力時間に対する影響が大きいことがわかる。シーケンシャル・ファイルでは，レコードの入力は順番に行われるので，１回のヒットミスでディスクキャッシュに読み出されるレコードの数が多い方が，当然ヒット率が高くなり入力時間は短縮される。

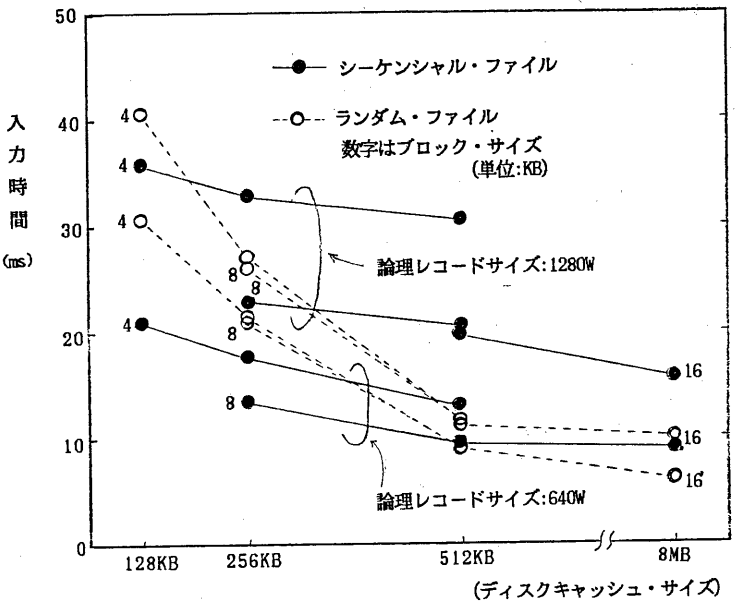

図７　ディスクキャッシュ・サイズと入力時間

ちなみに，論理レコード・サイズが１２８０Ｗの場合のディスクキャッシュ・サイズとヒット率の関係を示すと図８のようになる。図中の数字はブロック・サイズ（単位：ＫＢ）である。これにより，シーケンシャル・ファイルにおけるヒット率は，ディスクキャッシュ・サイズよりもブロック・サイズによる影響の方が大きいことが明確となる。また，ランダム・ファイルではブロック・サイズによる影響はほとんどなく，ディスクキャッシュ・サイズによる影響の方が大きいこともわかる。

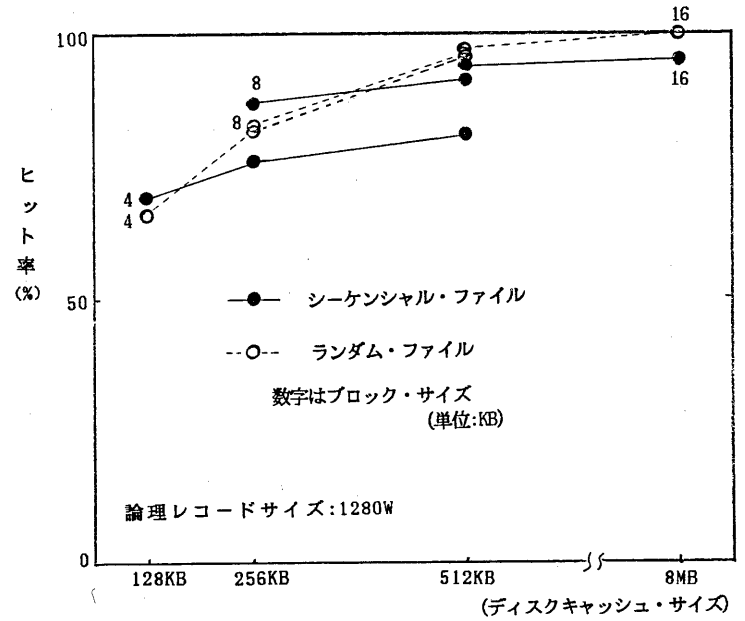

図８　ディスクキャッシュ・サイズとヒット率

４．３　ヒット率と入力時間

図９は，シーケンシャル・ファイル，ランダム・ファイルとも論理レコード・サイズが１２８０Ｗの場合のヒット率と入力時間の関係を示したものである。これによると，シーケンシャル・ファイル，ランダム・ファイルともヒット率が７５％から９５％で，入力時間はディスクキャッシュを使わない場合の約１／３から１／６に短縮されることがわかる。

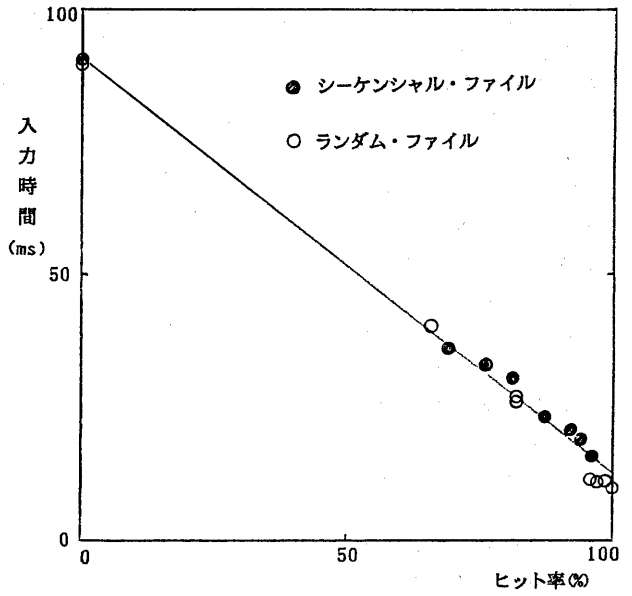

図９　ヒット率と入力時間

４．４　使用ＣＰＵ時間に対する影響

表２は，ディスクキャッシュを使用した場合と使用しない場合の１入出力の平均使用ＣＰＵ時間を示したものである。これによると，いずれの場合もディスクキャッシュを使用する方が，使用しない時よりもＣＰＵ時間を若干多く使うことがわかる。これは，ディスクキャッシュを使うときの入出力・ルーチンが異なるためである。

表２　使用ＣＰＵ時間に対する影響

（１入出力平均，単位ms）

アクセス方式 / ディスクキャッシュ	シーケンシャル		ランダム	
	入力	出力	入力	出力
使用せず	3.976	3.054	3.581	3.698
使　　用	4.030	4.090	3.652	3.724

５．おわりに

今回の計測によって，ディスクキャッシュのヒット率が７５％から９５％で，ディスク装置に対する入力時間が約１／３から１／６に短縮されることがわかった。ところで，通常の運用時のディスクキャッシュのヒット率をランダム・ファイルとシーケンシャル・ファイルについて調べてみるとそれぞれ約９７％と９１％になっている。したがって，通常の運用でもディスクキャッシュを使うプログラムの入力時間は，かなり短縮されているものと推測される。現在，ディスクキャッシュはコンパイラのワークファイルと入出力ファイル，システム・ライブラリ・ファイルおよびすべての用途のシーケンシャル・ファイルのアクセスのときしか使っていないが，今後はデータベースのアクセスへの適用などについても検討する予定である。

最後に，今回の実験計測にあたって，種々の御協力を戴いた日本電気（株）の大友裕氏と，発表の機械を与えて戴いた東北大学大型計算機センターの野口正一センター長に深謝する。

〈参考文献〉
（１）松沢，小畑，宮崎，三田，今井，表：ディスクキャッシュの効果に関する一考察，情報処理学会第２５回全国大会論文集，ｐｐ．１３３（１９８２）．
（２）小畑，松沢，宮崎，三田，今井，表：ディスクキャッシュの効果に関する一考察（その２），情報処理学会第２６回全国大会論文集，ｐｐ．１７５（１９８３）．
（３）小畑，松沢．宮崎，今井，表，神山：ディスクキャッシュの効果に関する一考察（その３），情報処理学会第２７回全国大会論文集，ｐｐ．１８７（１９８３）．
（４）平野正信：アクセス・ギャップを埋めるディスクキャッシュの機能を見る，日経コンピュータ，ｐｐ．７１－８５，（１９８３，３．２２）．
（５）猪苗代，徳永，平井，山本，石森：統合ディスクキャッシュ・システム，日本電気技報　Ｎｏ．１３３，ｐｐ．４５－５０（１９８０）．
（６）小畑，松沢，宮崎，三田，今井，表：統合ディスクキャッシュ・システムの方式と性能評価，全ＮＥＡＣユーザ会第７会シンポジウム論文集，ｐｐ．１３３（１９８３）．