負荷適応型ディスクキャッシュ制御機能の一方式の提案とその評価

井上太郎† 新村義章††

ディスクキャッシュは，ディスク制御装置（DKC）内に装備される半導体メモリであり，ディスク上のデータの一部分をキャッシングすることにより，主記憶装置とディスク装置の間のアクセス時間のギャップを埋めるものである．しかし，ディスクキャッシュはディスクと比べると小容量であるので，状況に応じて格納するデータを適切に選択しなければ性能が低下する場合がある．従来，ディスクキャッシュに格納するデータの選択は固定的であり，その変更のためにはユティリティプログラムを用いる必要があった．本論文では，状況に応じてディスクキャッシュに格納するデータの選択を動的に制御する負荷適応型ディスクキャッシュ制御機能を提案・評価した．本機能は，ファイルにディスクキャッシュの利用に関する優先度を設け，ディスクキャッシュ全体のヒット率の低下傾向時には，低優先度ファイルのディスクキャッシュの利用を禁止することにより，高優先度ファイルのヒット率の低下を防ぐ．本機能を試作し，ベンチマークプログラムによる実測の結果，本機能の適用によるスループットの低下はなく，高優先度ファイルのヒット率はランダムアクセスファイルの場合に向上することを確認した．このベンチマークプログラムは，実際の利用環境でのモニタデータから，ファイルアクセスの局所性の性質を抽出したモデルに基づいて，ファイルへのアクセスを発生するプログラムである．

A Proposal of Disk-Cache Control Function Adapting to Load and Its Evaluation

TARO INOUE† and YOSHIAKI SHINMURA††

A disk-cache is semiconductor memory equipped in a disk controller (DKC). It covers the access time gap between main storage and disk unit, by caching the part of data stored in the disk unit. As the capacity of disk-cache is smaller than that of disk unit, the performance of the system may decrease, unless the selection of cached data is adequate. As the selection of cached data is not dynamic, we must use the utility program in modifying it. We proposed and evaluated a disk-cache control function adapting to load. The user defines a priority for the use of a disk-cache. When the hit ratio of a disk-cache is tend to decrease, the low priority files are inhibited the use of a disk-cache, and it prevents from dropping the hit ratio in the high priority files. We evaluated this function by using a benchmark program, and verified that the throughput of the system didn't drop, and the hit ratio of high priority files increased on random access files. The benchmark program generates the accesses to the files, based on the model which extract the character of a locality in a file access from the monitoring data in real environment.

1. はじめに

1.1 ディスクキャッシュの概要

（1）記憶階層でのディスクキャッシュの位置付け

コンピュータシステムにおいては，コスト，アクセス速度，記憶容量の関係に従って記憶階層が構成される．この階層の中で，主記憶装置（Main Storage: MS）とディスク装置（Disk Unit: DKU）との間のアクセス時間のギャップの大きさが重大な問題となっている．このギャップは，ディスク装置にシーク時間や回転待ち時間などの機械的な動作が存在することに起因する．このギャップを埋めるものとしてディスクキャッシュ[1)]がある．ディスクキャッシュは，ディスク制御装置（Disk Controller: DKC）内に装備される半導体メモリで，ディスク上のデータの一部分が格納（キャッシング）される．ディスクキャッシュを装備するディスク制御装置のことをキャッシュ付きディスク制御装置（Cached DKC）という．

（2）ディスクキャッシュの動作と動作モード

中央処理装置（Central Processing Unit: CPU）

† (株)日立製作所システム開発研究所
System Development Laboratory, Hitachi, Ltd.
†† (株)日立製作所ソフトウェア開発本部
Software Development Center, Hitachi, Ltd.

がディスク装置上のレコードをリードしようとするとき，このレコードがディスクキャッシュ上に存在すれば（これをリードヒットという），ディスクキャッシュからレコードを読み込むので，上記の機械的な動作を行わずにリードできアクセス時間を短縮できる．一方，レコードがディスクキャッシュ上に存在しなければ（これをリードミスという），ディスク装置からCPU へレコードを転送してから，このレコードを含むトラック全体をディスクキャッシュに読み込む．トラックスロット（ディスクキャッシュ上の1トラックに相当する領域）は LRU (Least Recently Used) 方式で管理される．

また，DKC におけるディスクキャッシュの管理方式としていくつかのモードがある[2]．その主なものを以下に示す．これらのモードは，入出力時に OS がチャネルに指示を送ることにより選択される．これらのモードを使い分けることにより，ディスクキャッシュの利用の許可/禁止を OS から制御することができる．

【ランダムアクセスモード】

リードミスの発生時には，当該レコードを含むトラックをディスクキャッシュにロードする．ランダムアクセスファイルに適用する．

【順次アクセスモード】

あるレコードが読み出されたとき，そのレコードを含むトラックから複数トラック先までのトラックがディスクキャッシュ上に存在しているか否かをチェックし，存在しなければディスクキャッシュにロードする．これにより後続のアクセスでのヒットが期待できる．順次アクセスファイルに適用する．

【キャッシュロード禁止モード】

リードキャッシュが発生しても，当該レコードを含むトラックをディスクキャッシュにロードせず，他のトラックをディスクキャッシュから追い出さない．よって，新たにディスクキャッシュを使用することがない．

1.2 ディスクキャッシュの利用上の問題点

ディスクキャッシュの容量はディスクの容量と比べると小さく，前記のようにその領域は LRU 方式で管理されるので，使用頻度の低いデータはディスクキャッシュから追い出される．したがって，状況に応じてディスクキャッシュに格納するデータを適切に選択することが必要である．すなわち，アクセスの特性に合わせて前記の各モードを適切に使い分けることが必要である．さもなければ，ディスクキャッシュを利用したためにかえって性能が低下するということがある．

従来は，OS がファイルの属性（ランダムアクセスファイル/順次アクセスファイル）に従って，ランダムアクセスモードあるいは順次アクセスモードを固定的に選択している．つまり，基本的には，すべてのファイルがディスクキャッシュの利用対象となっているのである．局所性が低く大量にディスクキャッシュを使用するようなファイルにランダムアクセスモードを適用すると，他のファイルがディスクキャッシュから追い出されるということが起こる．このような性質の悪いファイルは，他のファイルに悪影響を及ぼさないようにするために，ディスクキャッシュの使用を禁止する方がよいといえる．

しかし，従来は，ディスクキャッシュの利用対象の選択は固定的であり，ディスクキャッシュの利用の許可/禁止を変更するには，ユーザがユティリティプログラムを利用するしかなく，これがディスクキャッシュの利用上のネックとなっていた．

そこで，文献1)ではディスクキャッシュの利用を動的に許可/禁止することを提案し，文献3)ではヒット率の敷居値（固定値）を設定して，それを境にディスクキャッシュの利用の許可/禁止を制御する方式を示している．しかし，その時々の状況（ディスクキャッシュの容量，負荷の性質等）により適切な敷居値は異なるはずであるので，文献3)に示されるような固定的な敷居値に基づく制御では，その敷居値をどのような値に設定すべきかという点が問題となる．したがって，その時々の状況の変動に適応して敷居値が変化することが望ましい．

本論文では，ディスクキャッシュ全体のヒット率およびその時間的な変化の傾向（上昇/安定/下降）に着目してヒット率の敷居値を動的に設定することにより，ディスクキャッシュの利用の許可/禁止を制御する機能—負荷適応型ディスクキャッシュ制御機能—を提案する．

1.3 ディスクキャッシュを有するシステムの評価

計算機システムの評価は，そのシステムが使用されるのと同様の環境で同様の負荷を与えて行うのが理想的である．文献1)では，現実のシステムでの入出力に関するトレースデータから負荷を生成して，シミュレーションによりディスクキャッシュの性能評価を行っている．しかし，シミュレーションではなく実測の

場合には，実際のシステムで使われるプログラムを同じ環境で動作させることは難しい．そこで，実際のシステムで使われるプログラムの動作の特徴を反映した人工的な負荷を用いて評価を行うということが考えられる．

ディスクキャッシュの性能はヒット率により左右される．そして，ディスクキャッシュの領域はLRU方式により管理されるので，ディスクキャッシュの性能はファイルアクセスの局所性に左右される部分が大きい．したがって，ディスクキャッシュを有するシステムの性能を評価するには，実際のシステムで使われるプログラムでのファイルアクセスの局所性の性質を反映した人工的な負荷を用いるのが望ましいといえる．

本論文では，第2章において，1.2節で述べた問題点を解決する負荷適応型ディスクキャッシュ制御機能の詳細を示し，第3章では，ディスクキャッシュを有するシステムの評価のためのベンチマークプログラムについて述べる．第4章では，シミュレーションによる効果の予測を示し，第5章では，ベンチマークプログラムを用いて負荷適応型ディスクキャッシュ制御機能の性能実測を行った結果について述べる．

2. 負荷適応型ディスクキャッシュ制御機能

2.1 機能の概要

1.2節で述べたように，負荷適応型ディスクキャッシュ制御機能では，その時々の状況の変動に適応したヒット率の敷居値を設定するために，ディスクキャッシュ全体のヒット率およびその時間的な変化の傾向(上昇/安定/下降)に着目する．このように，ヒット率の値そのものだけでなく，その時間的な変化の傾向を考慮することにより，ディスクキャッシュへの負荷の変動を検知することを可能とする．

そして，ファイルに関してディスクキャッシュの使用に関する優先度の考え方を導入する．すなわち，

- 高優先度（USE）：ディスクキャッシュを必ず使用する
- 低優先度（AUTO）：ディスクキャッシュを使用するか否かはシステムが決定する

とする．これにより，局所性が低く大量にディスクキャッシュを使用するようなファイルのディスクキャッシュの使用の禁止を可能とし，他のファイルへ悪影響を及ぼさないようにする．

負荷適応型ディスクキャッシュ制御機能の概要を**図1**に示す．

（a）ディスクキャッシュの使用に関する優先度（高優先度（USE）：ディスクキャッシュを必ず使用する，低優先度（AUTO）：ディスクキャッシュを使用するか否かはシステムが決定する）を，各ファイルの重要度やアクセス性能に対する要求に応じてユーザが定義する．局所性が低いファイルには低優先度（AUTO）を指定するべきである．

（b）ディスクキャッシュ全体のヒット率をモニタする．

（c）モニタの結果，ディスクキャッシュ全体のヒット率が安定／上昇傾向にある場合，すべてのファイルにディスクキャッシュの使用を許可する．

（d）モニタの結果，ディスクキャッシュ全体のヒット率が下降傾向にある場合，高優先度（USE）ファイルにのみディスクキャッシュの使用を許可して，低優先度（AUTO）ファイルにはディスクキャッシュの使用を禁止する．

（e）（c）あるいは（d）での判定に基づきキャッシュモードをOSへ指示する．すなわち，ディスクキャッシュの利用を許可する場合には，ランダムアクセスモードあるいは順次アクセスモードをファイルの属性に従って指示し，ディスクキャッシュの利用を禁止する場合には，キャッシ

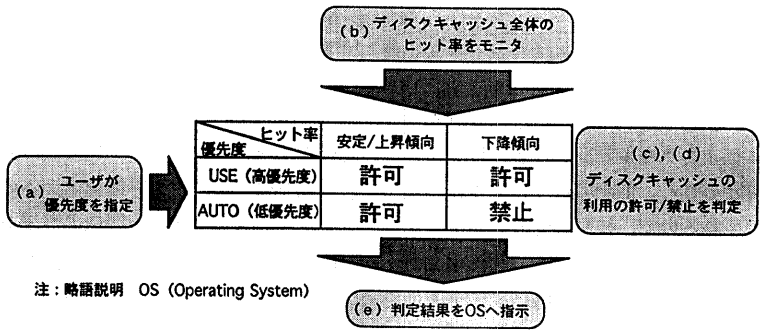


図1 負荷適応型ディスクキャッシュ制御機能の概要
Fig. 1 Aspect of disk-cache control function adapting to load.

ュロード禁止モードを指示する．

2.2 制御方式の詳細

（1）ヒット率の変化の傾向の判定と制御の開始

ディスクキャッシュ全体でのヒット率の変化の傾向の判定には，過去n回のモニタにおけるディスクキャッシュ全体でのランダムアクセスモードのリードヒット率の平均値を用いる．これは，極端なリードヒット率の変動の影響を排除して，ヒット率の変動の傾向を知るためである．時刻 $t_i, t_{i-1}, t_{i-2}, \cdots$ においてモニタを行うとするとき，

$HR(t_i)$：時刻 t_{i-1} と時刻 t_i の間でのディスクキャッシュ全体でのランダムアクセスモードのリードヒット率

$$m(t_i, n)=(1/n)\sum_{j=1}^{n} HR(t_{i-j+1})$$

とする．ここで，時刻 t_i において以下の条件が成立した場合に，ヒット率が下降傾向であると判定して制御を開始し，低優先度（AUTO）ファイルのディスクキャッシュの使用を禁止する．

$$HR(t_i)\leqq\alpha\times m(t_i, n) \quad (0\leqq\alpha\leqq 1).$$

α はヒット率の下降度を示すパラメータである．

（2）制御の終了

時刻 t_{i+K} において以下の条件が成立した場合に，制御を終了し，低優先度（AUTO）ファイルにディスクキャッシュの使用を禁止することをやめる．

$$HR(t_{i+K})>\alpha\times m(t_i, n).$$

すなわち，ヒット率が制御開始時のヒット率の限界値（$=\alpha\times m(t_i, n)$）まで回復したときに制御を終了する．ただし，上記のKの値には上限値を設け，その上限値を越えてもヒット率が回復しない場合には強制的に制御を終了し，低優先度（AUTO）ファイルにディスクキャッシュの使用を禁止することをやめる．

3. ディスクキャッシュシステム向けベンチマークプログラム

1.3 節で述べたように，ディスクキャッシュを有するシステムの性能を評価するには，実際のシステムで使われるプログラムでのファイルアクセスの局所性の性質を反映した人工的な負荷（ベンチマークプログラム）を用いるのが望ましい．

そこで，以下の手法により，負荷適応型ディスクキャッシュ制御機能の評価のためのベンチマークプログラムを開発した．本ベンチマークプログラムは現実のシステムで使われるプログラムのファイルアクセスの局所性の性質を反映したものである．以下では，時間に関する局所性および空間に関する局所性の2つに分けてファイルアクセスのモデル化を考える．

3.1 時間に関する局所性のモデル化

（1）ファイルの分類

現実のシステムでのモニタリングデータから，時間の経過に伴う各ファイルへのアクセス回数の分布を調べたところ，ファイルは大きく次の2種類に分類できる（**図2**参照）．

（a）一様型ファイル

アクセスが全時刻に渡って平均的に発生するファイル．時間に関する局所性が低いファイルである．

（b）局所型ファイル

アクセスが特定時刻の周辺に集中して発生するファイル．時間に関する局所性が高いファイルである．局所型ファイルでは，個々のアクセスを見るのではなく，集中した発生したアクセスの塊に注目し，これを「アクセス群」としてとらえる．

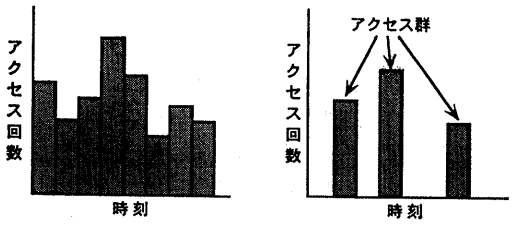
ただし，あるファイルが一様型ファイルおよび局所型ファイルのどちらに属するのかを判定する基準は，全時間区間に対するアクセス群が発生する時間区間の割合が0.5以上なら一様型ファイル，0.5未満なら局所型ファイルとする．この基準に基づいて，システムのすべてのファイルを分類する．

（2）アクセス発生時刻のモデル化

（a）一様型ファイル

全時刻に渡って平均的にアクセスが発生しているので，アクセス発生間隔が正規分布に従うと仮定してモデル化する．

すなわち，モニタリングデータから一様型ファイルでのアクセス発生間隔の平均および分散を算出し，ベンチマークプログラムにおいては，この平均および分散を有する正規分布に従ったアクセス発生間隔でアクセスを発生させる．



（1） 一様型ファイル （2） 局所型ファイル

図2 一様型ファイルと局所型ファイル
Fig. 2 Uniform type file and locality type file.

表 1 モニタ結果とベンチマークプログラムの実行結果の比較
Table 1 Comparison the result of monitoring to the result of benchmark program execution.

項目		ファイル編成 順編成	区分編成	直接編成	仮想記憶編成 キー順ファイル INDEX	キー順ファイル DATA	その他
平均アクティブトラック数		66.5	2.1	35.0	1.0	10.3	88.8
(/ファイル)		43.8	3.0	35.0	1.0	7.5	73.6
一様型ファイル	ファイル数	0	2	0	0	1	0
		0	2	0	0	1	0
	1分間の平均アクセス回数(回)	—	43.3	—	—	11.1	—
		—	35.6	—	—	8.8	—
	λの平均	—	0.841	—	—	0.724	—
		—	0.841	—	—	0.724	—
	平均アクセス間隔(秒)	—	2.67	—	—	5.34	—
		—	2.92	—	—	5.94	—
局所型ファイル	ファイル数	8	5	12	25	131	104
		8	5	12	35	131	104
	1分間の平均アクセス回数(回)	23.1	1.7	58.4	2.8	47.4	367.3
		18.0	0.9	58.4	1.9	34.2	340.1
	λの平均	0.018	0.039	0.049	0.090	0.095	0.120
		0.027	0.035	0.049	0.090	0.071	0.097
	平均アクセス群数(個)	4.3	9.2	11.7	21.4	22.6	27.6
		1.9	2.4	11.7	21.4	4.9	6.7
	アクセス群当たりの平均アクセス回数(回)	162.5	8.8	99.8	1.3	3.8	30.6
		83.1	5.3	99.8	1.3	3.7	33.9

本表の各項目において，上段は実システムのモニタ結果を，下段はベンチマークプログラムの実行結果を示す．

(b) 局所型ファイル

個々のアクセスではなく，アクセス群の発生の分布のモデル化を考えることとし，これをポアソン分布でモデル化する．ポアソン分布 $P_\lambda(r)$ は，

$$P_\lambda(r)=e^{-\lambda}\cdot\lambda/r!$$

で定義される0および自然数上の分布であり，「1区画あたり平均 λ 個の割合で粒子を落としたとき，1区画に実際に r 個の粒子が落ちる確率が $P_\lambda(r)$ である」と解釈できる．そこで，アクセス群は全時間区間に渡って均等に発生するものとして，

λ=(アクセス群が発生していた時間区間数)/(総時間区間数)

とし，各時間区間を上記の区間と考えると，ある時間区間においてアクセス群が r 個発生する確率は $P_\lambda(r)$ となる．

モニタリングデータからは，時間区間ごとのアクセスの発生の有無を調べ，アクセスがあった場合には当該時間区間にはアクセス群が発生しているものとみなして局所型ファイルにおける λ の平均値を算出する．ベンチマークプログラムにおいては，この λ を有するポアソン分布に従って時間区間ごとにアクセス群を発生させる．ただし，1つのアクセス群の中で発生させるアクセスの回数は，モニタリングデータから算出したアクセス群当たりの平均アクセス回数である．

3.2 空間に関する局所性のモデル化

モニタリングデータから各トラックへのアクセス回数を調べ，少なくとも1回はアクセスのあったトラックをアクティブトラックと呼ぶ．個々のアクセスは，アクティブトラックのすべてに渡って均等にアクセスが発生するものとし，各アクティブトラックへのアクセス回数の分布を

λ=(総アクセス回数)/(アクティブトラック数)

のポアソン分布でモデル化した．

以上に述べた手法により作成したベンチマークプログラムを実行した結果と実システムのモニタリング結果の比較を**表1**に示す．ここでは，ファイルをファイル編成ごとに分け，ファイル編成ごとに各項目の平均値を算出した．

4. シミュレーションによる効果の予測

負荷適応型ディスクキャッシュ制御機能の効果を予測するために，前記のベンチマークプログラムに用いたモニタリングデータに記録されたアクセスを負荷として，ディスクキャッシュの動作のシミュレーション

を行った．ディスクキャッシュの容量は 32M バイトとしている．

ただし，このシミュレーションは，AUTO を指定した低優先度ファイルのディスクキャッシュの使用を禁止した場合のシステムの挙動を知るために行ったものであり，低優先度ファイルはシミュレーション期間中は常にディスクキャッシュの使用を禁止しており，ディスクキャッシュの使用の許可/禁止を動的に制御してはいない．よって，3章で示した負荷適応型ディスクキャッシュ制御機能のアルゴリズムをそのまま忠実にシミュレートしたものではない．

シミュレーション結果を**図3**に示す．ここでは，低優先度 (AUTO) を指定してディスクキャッシュの使用を禁止するファイルを変化させ，それを「ディスクキャッシュの使用を禁止した I/O の割合」として横軸に示している．ディスクキャッシュの使用を禁止した I/O の割合が 0% ということは，すべてのファイルに高優先度 (USE) を指定してディスクキャッシュの使用を許可していることになる．

このシミュレーション結果より，

- ディスクキャッシュの使用を禁止した I/O の割合が 10% 程度までは，すべてのファイルに高優先度 (USE) を指定してディスクキャッシュの使用を許可している場合と比べてヒット率が低下しない（システムのスループットが低下しない）．
- ディスクキャッシュの使用を禁止した I/O の割合が増加するにつれて，高優先度 (USE) を指定してディスクキャッシュの使用を許可されたファイルのヒット率は向上する．

ことがわかる．

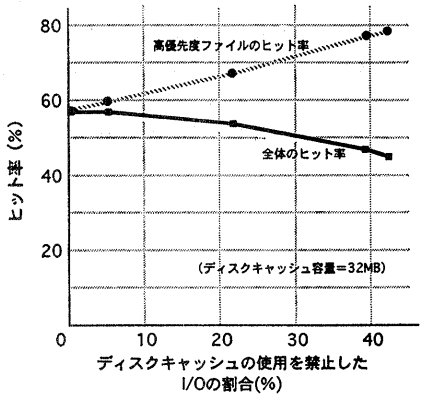


図 3　シミュレーション結果
Fig. 3　Result of simulation.

5. 実測による評価

5.1 測定環境

負荷適応型ディスクキャッシュ制御機能を汎用大型 OS 上に試作し，ボリューム単位に優先度 (USE/AUTO) を指定して，前記のベンチマークプログラムを用いて性能評価を実施した．実測は 32M バイトのディスクキャッシュを装備したディスク制御装置 (DKC) に8ボリュームを接続したシステムで行った．ただし，$\alpha=0.9$，$n=5$ に設定した．

5.2 測定結果

（1）全ボリュームのヒット率（**図4**）

従来方式と提案方式との全ボリュームでのヒット率の比較を図4に示す．ここでは，USE および AUTO を指定したボリュームを含むすべてのボリュームにおけるヒット率を比較している．この比較により，提案方式の適用によるシステムのスループットの変化がわかる．各ケースともヒット率の変化はほとんどなく，負荷適応型ディスクキャッシュ制御機能の適用によるスループットの低下はないといえる．

（2）高優先度ボリュームのヒット率（**図5**）

従来方式と提案方式との高優先度ボリュームのヒット率の比較を図5に示す．高優先度ボリュームのヒット率は，ランダムアクセスファイルにおいては提案方式の方が従来方式よりも 4% 程度向上する．

また順次アクセスファイルにおいては，本方式を適用してもほとんど変化がない．これは，順次アクセスファイルでは次にアクセスするデータをディスクキャッシュへ先読みするため，そのヒット率はほぼ 100% に近く，それ以上にヒット率が向上する余地がないためである．

従来方式と提案方式との全ボリュームでのヒット率の比較においてヒット率の変化はほとんどなく，高優先度ボリュームのヒット率は提案方式の方が従来方式

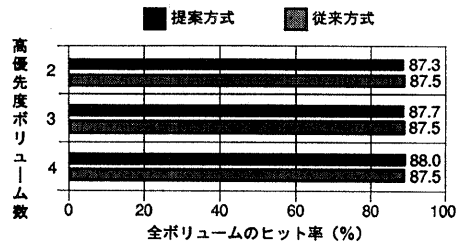


図 4　全ボリュームのヒット率の比較
Fig. 4　Comparison of hit ratio of whole volume.

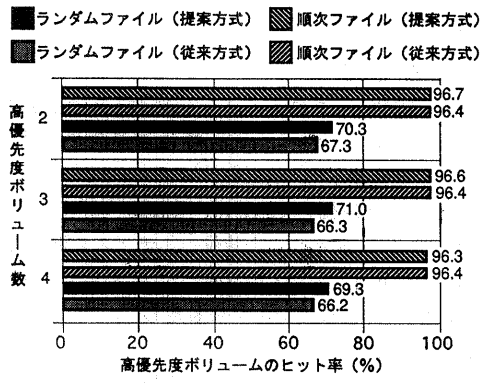


図 5 高優先度ボリュームのヒット率の比較
Fig. 5 Comparison of hit ratio of high priority volume.

よりも向上している．これは，提案方式の適用により，低優先度ボリュームのディスクキャッシュの使用が禁止された結果，低優先度ボリューム上のデータが高優先度ボリューム上のデータをディスクキャッシュから追い出すことが減り，高優先度ボリューム上のデータがディスクキャッシュに存在する確率が高くなることによるものと考えられる．

6. おわりに

ディスクキャッシュの負荷状況に応じて，ディスクキャッシュの利用対象の選択を動的に制御する負荷適応型ディスクキャッシュ制御機能を提案した．本機能を試作し，ベンチマークプログラムによる実測の結果，スループットの低下はなく，高優先度を指定したファイルのヒット率はランダムアクセスファイルの場合に向上することを確認した．また，評価に用いたベンチマークプログラムについても述べた．

謝辞 本研究の推進にあたり，(株)日立製作所システム開発研究所堂免信義元所長のご指導に感謝する．元システム開発研究所梅野英典主任研究員ならびに金居貞三郎研究員には多くの議論をして頂いた．システム開発研究所新井利明主任研究員にも多くの助言を頂いた．また，(株)日立製作所ソフトウェア開発本部今居和男主任技師には貴重なご意見を頂いた．(株)日立製作所ソフトウェア開発本部ならびに(株)日立ソフトウェアエンジニアリングの方々からは，多くのご意見とご協力を頂いた．ここに感謝の意を表す．

参考文献

1) Smith A. J.: Disk Cache-Miss Ratio Analyis and Design Considerations, *ACM Trans. on Computer Systems*, Vol. 3, No. 3, pp. 161-203 (1985).
2) 日立製作所：HITAC VOS3/AS システムユティリティ/独立ユティリティ，6180-3-224-40, pp. 229, 日立製作所 (1993).
3) Montgomery, C. A.: Dynamic Cache Management, *Proc. of the CMG '90* (Dec. 1990, Orlando), pp. 662-670 (1990).

(平成5年12月16日受付)
(平成6年 2 月17日採録)

井上 太郎（正会員）
1961 年生．1984 年京都工芸繊維大学工芸学部生産機械工学科卒業．1986 年同大学院修士課程修了．同年，(株)日立製作所入社．現在，同社システム開発研究所に勤務．仮想計算機システム，ストレージ管理システム，分散システムの研究開発に従事する．システム制御情報学会会員．

新村 義章（正会員）
昭和 28 年生．昭和 49 年鹿児島工業高等専門学校電気工学科卒業．同年(株)日立製作所入社．現在，同社ソフトウェア開発本部主任技師．大型オペレーティングシステムの統合ストレージ管理システムの研究に従事．