

多段キャッシュ環境の下位キャッシュ における参照の時間的局所性の解析

竹内 洸祐[†] 長廻 雄介[†] 山口 実靖[†]

多段キャッシュ環境において上位キャッシュの置換アルゴリズムに LRU が使用されている場合、下位キャッシュにおいては一度アクセスされたデータが近い将来に再度アクセスされる可能性が低くなり、通常とは逆向きの負の参照の時間的局所性が存在する事がシミュレーションにより確認されている。

本研究では実計算機上で、実アプリケーションを動作させ、下位キャッシュへのアクセスパターンの解析を行うことにより、負の参照の時間的局所性の存在の調査を行った。

Analyses of Locality of Reference in L2 Cache

KOSUKE TAKEUCHI[†] YUSUKE NAGASAKO[†]
SANEYASU YAMAGUCHI[†]

In network storage environment, two caches are used. They are server computer cache and storage appliance cache. In many cases, LRU cache replacement algorithm is applied in server computers. Applying LRU in the upper cache leads negative temporal locality of reference in the lower cache. Thus LRU does not work effectively in the lower cache. In this paper, we investigate locality of reference in the lower cache and performance of existing cache replacement algorithm.

1. はじめに

近年、ストレージ容量は急速に増大を続け、計算機が扱うデータの量は増加の一途を辿っている。それに伴うストレージ管理コストの増加が情報社会の問題の一つとなっている。そこで SAN(Storage Area Network)や NAS(Network Attached Storage)を用いてネットワーク上にストレージを配置し、データを集約する手法が提案された。通常のローカル接続のストレージに対して、これらのネットワーク接続のストレージはネットワークストレージと呼ばれる。ユーザが使用するデータを集約し一元管理することによりストレージ管理コストの大幅な削減が可能であり、ネットワークストレージは多くの組織で使用されるようになった。

ネットワークストレージ上のデータへのアクセスは、サーバー計算機上のキャッシュとストレージ機器上のキャッシュの二重のキャッシュを介して行われる。このような環境の場合、両キャッシュには同一のデータを重複して保存する可能性がある。たとえば、サーバー計算機キャッシュでミスヒットとなったアクセス要求はストレージ機器に転送され、ストレージ機器キャッシュでもミスヒットとなった場合はそのアクセスデータは最新のデータとして、ネットワークストレージキャッシュとサーバー計算機キャッシュの両方に重複して格納される。両キャッシュの保持データの多くが重複している場合、サーバーキャッシュでミスヒットしたアクセス要求は高確率でストレージ機器上のキャッシュでもミスヒットし、ストレージ機器のキャッシュは効果的に機能しない。同様にネットワークストレージキャッシュに格納されているデータへのアクセスは、サーバー計算機キャッシュにおいてキャッシュヒットとなり、ネットワークストレージへ転送されず、ネットワークストレージのキャッシュが効果的に機能しない。これをストレージ機器の視点で見ると、ひとたびアクセスされたデータは(サーバーキャッシュに保持され以後サーバー内で処理されるため)近い将来に再度アクセスされる可能性が低いということになる。すなわち、通常とは逆向きの負の参照の時間的局所性が存在することとなる。これに関連する既存の研究として、長廻らのシミュレーションによる解析[1]や、Zhou らによる DBMS や DISK I/O 用いた解析[2]があるが、共に実システムの動作を十分に考慮しているとは言えず、実システム上で実アプリケーションを動作させての解析が必要であると考えられる。

本論文では多段キャッシュ環境における負の参照の時間的局所性について調査を行うため、実計算機上で実アプリケーションを動作させ、下位キャッシュへのアクセスパターンの解析を行い、負の参照の時間的局所性の存在の調査を行う。そして、そのアクセスログを用いて既存キャッシュ置換アルゴリズムのキャッシュヒット率の評価

[†] 工学院大学院工学研究科電気・電子工学専攻
Electrical Engineering and Electronics, Kogakuin University Graduate School

を行う。

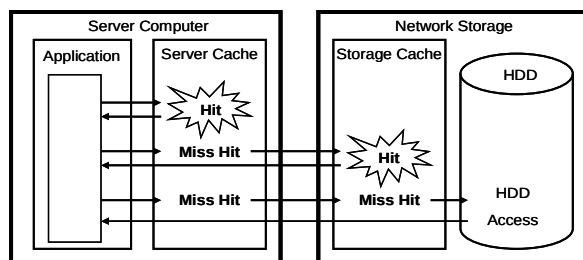


図 1. 二重キャッシュ構造

2. 二重キャッシュ構造

図 1 は二重キャッシュ環境における両キャッシュの動作とアクセス要求の転送の様子を示したものである。サーバー内のアプリケーションからストレージ機器上のデータにアクセス要求が発行されると、まず該当データがサーバーキャッシュ内に存在するか否かが確認される。存在する場合は、サーバー計算機キャッシュヒットとなり、サーバーキャッシュによりアプリケーションにデータが返される。該当データが存在しない場合、データアクセス要求はネットワークストレージに転送され、ネットワークストレージキャッシュに該当データが存在するか否かが確認される。該当データが存在する場合は、ストレージ機器のキャッシュでデータがサーバー計算機に返され、HDD アクセスは行われない。ストレージ機器上のキャッシュにも該当データがない場合にはストレージキャッシュもミスヒットとなり、ストレージ機器上の HDD にアクセスを行い、該当データをアプリケーションに返す。HDD へのアクセスはキャッシュ（半導体メモリ）へのアクセスに比べアクセス時間が長く、ネットワークを介した場合はその遅延により、アプリケーションにデータが返るまでにかかる時間が増える。よって、アクセス時間は短い順に、サーバー計算機キャッシュヒット、ストレージ機器キャッシュヒット、HDD デバイスアクセスとなる。

サーバー計算機キャッシュがミスヒットした場合(ストレージキャッシュヒットや HDD アクセス)の場合、アクセス対象データはストレージキャッシュに最新データとして格納され、同時にサーバー計算機キャッシュにも最新データとして格納される。このような処理により、両キャッシュには同一のデータが格納されてしまう可能性がある。両キャッシュに同一のデータが重複して格納され両キャッシュが保持しているデータが類似している場合は、ネットワークストレージキャッシュが保持しているデータへのアクセス要求はサーバー計算機キャッシュにより処理されてしまい、ネットワークストレージキャッシュが効果的に機能しない。逆に、サーバーキャッシュでミス

ヒットとなったアクセス要求のデータは、ネットワークストレージキャッシュも保持しておらずネットワークストレージキャッシュにおいてもミスヒットとなる。このため両キャッシュの内容の多くが重複している場合、ネットワークストレージのキャッシュは効果的に機能せず、両キャッシュの重複は避けることが好ましいと言える。

3. 負の参照の時間的局所性

3.1 参照の局所性

計算機システムのメモリ階層では、キャッシュには下層の記憶装置と比較して小さな容量の記憶装置しか用意されないが、多くの場合十分な性能を発揮できることが経験的に知られている。これはアプリケーションが発行する要求には多くの場合偏りがあるためであり、この参照の偏りは参照の局所性(Locality of Reference)と呼ばれている。現在のメモリ管理技法の多くはこの参照の局所性の概念に基づいている。理論的に証明された性質ではなく観察により得られた経験則であり、この性質を持たないアクセス要求を発行するアプリケーションも存在する。

参照の局所性には、“時間的局所性 (Temporal locality)” と、空間的局所性 (Spatial locality)”がある。時間的局所性は、ある時点で参照されたデータが近い将来にも再び参照される可能性が高いという性質である。空間的局所性は、あるデータが参照されたとき、その近傍のデータが近い将来に参照される可能性が高いという性質である。

3.2 LRU

上記の参照の局所性を期待したキャッシュ置換手法に LRU(Least Recently Used)がある。LRU は、最近アクセスされたデータ(最後のアクセスからの時間が最も短いデータ)が再アクセスされる確率が最も高く、最後のアクセスからの時間が最も長いデータが再アクセス確率が最も低いと仮定し、最後のアクセスからの時間が最長であるものを破棄するキャッシュ置換手法である。

多くのコンピュータシステムのアプリケーションにおいて参照の時間的局所性が存在することが確認されており、現在の OS やコンピュータシステムのほとんどにおいてキャッシュの置換アルゴリズムとして LRU が採用されている。

3.3 負の参照の時間的局所性

一般的に、サーバー計算機のキャッシュ置換アルゴリズムには LRU が使われている。この場合サーバー計算機上のキャッシュには、最近アクセスされたデータが格納されている。よって、アプリケーションからの「最近アクセスされたデータ」へのアクセスはサーバー計算機キャッシュにより処理され、ネットワークストレージには転送されない。よって、ネットワークストレージキャッシュには「最近アクセスされたデー

タ」へのアクセス要求が到達せず、最近アクセスされたデータが近い将来に再度アクセスされる可能性は低いという、通常とは逆向きの負の参照の局所性が存在することになる。よって、通常の参照の時間的局所性を期待している LRU をネットワークストレージキャッシュにおいて用いても、効果的には機能しないと予想される。

図 2 に、シミュレーションにより得たサーバー計算機キャッシュとストレージ機器キャッシュにおける再アクセス時間と再アクセス確率の関係を示す[1]。再アクセス時間が 0[アクセス]～約 1000[アクセス]のときはネットワークストレージにアクセス要求が来ず、ネットワークストレージキャッシュにおける再アクセス確率が 0[%]になっていることがわかる。これより、シミュレーション環境においては明確に負の参照の局所性が存在することがわかる。しかし、本シミュレーションは実環境で用いられるファイルシステムなどの振る舞いが考慮されておらず、実際の OS を実計算機上で稼働させ、その上で実アプリケーションを動作させた実環境における考察が必要であると考えられる。

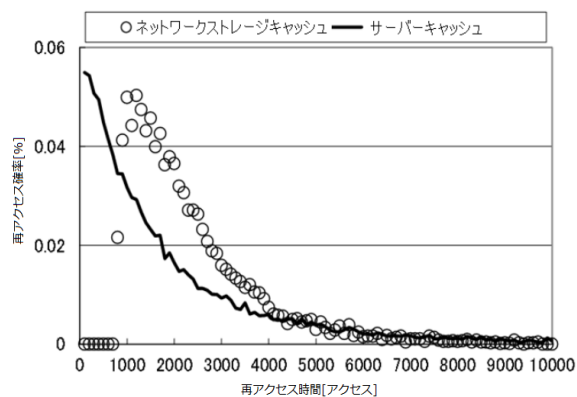


図 2. アクセス経過時間とアクセス確率の関係

4. 下位キャッシュにおける参照の時間的局所性の解析

4.1 解析方法

本章において、実環境のキャッシュに対する参照の局所性の調査を行う。図 3 の様なネットワークストレージ環境を想定し、図内の“Monitoring”の箇所における参照の局所性の調査を行う。図内の“Monitoring”の箇所を流れるアクセス要求は、サーバー計算機 OS のキャッシュを経由した(ミスヒットした)後のものであり、下位キャッシュへの参照の局所性を調査することができる。本研究では図 3 の環境を想定して、図 4 の実験環境を構築して“Monitoring”の箇所にてアクセスログの収集を行った。

図 4 の実験環境にてベンチマークソフトである FFSB(Flexible File System Benchmark)を実行した。なおモニタリングは OS の SCSI 層にて行った。実験に使用した計算機の仕様は表 1 の通り、FFSB の設定は表 2 の通りである。

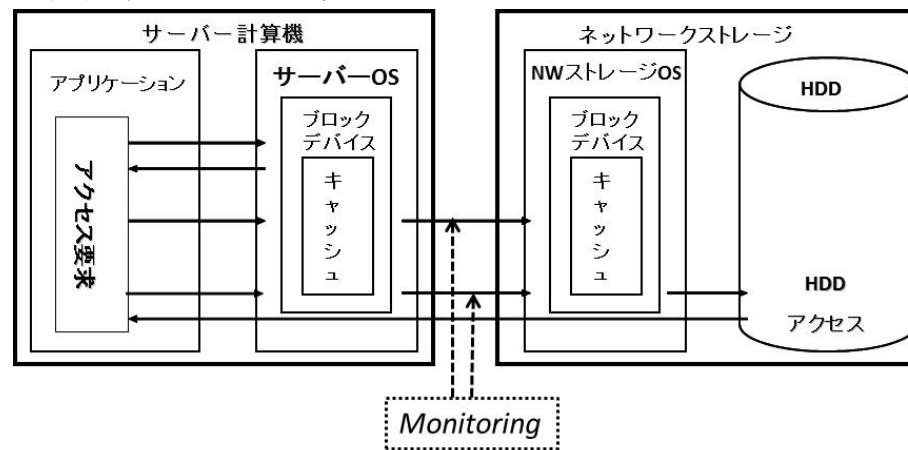


図 3. 想定環境

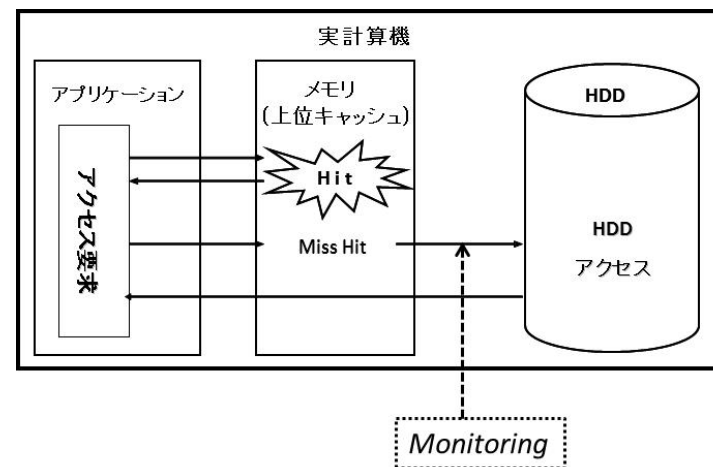


図 4. 測定環境

表 1.実計算機の仕様

OS	CentOS5.5(64bit)
Kernel	Linux 2.6.18.8
CPU	AMD Athlon™
HDD(sda)	140GB
HDD(sdb)	80GB
メモリ	4GB
Cache	3.46GB
使用ファイルシステム	ext2,ext3,NTFS

表 2.FFSB の設定

Data Size	16GB	16GB	16GB	16GB
File Size	16MB	16MB	16MB	16MB
Num File	1024	1024	1024	1024
OP Size	16KB	64KB	256KB	1MB

4.2 測定結果

再アクセス間隔とアクセス発生確率(確率密度関数)の関係を図 5~8 に示す。

図 5 より,ext3 では再アクセス間隔 3.1[GB]以下の再アクセス確率が, NTFS では 1.6 [GB]以下の再アクセス確率が極めて低く負の参照の時間的局所性が存在することが確認された。同様に図 6 の ext3 の再アクセス間隔 2.7[GB]以下, NTFS の 1.5[GB]以下, 図 7 の ext2 の 3.2[GB]以下, ext3 の 3.2[GB]以下, NTFS の 1.7[GB]以下にて再アクセス確率が極めて低いことが確認された。以上より, ファイルシステムに ext3, NTFS を用いた場合は負の参照の時間的局所性が存在し, ext2 を用いた場合は必ずしも存在しないことが確認された。

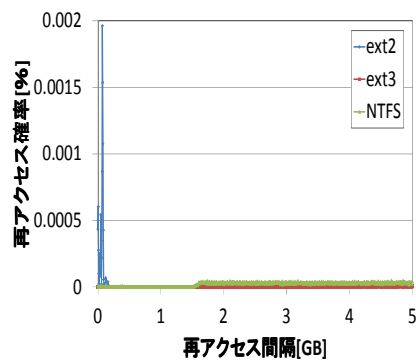


図 5. FFSB (OPsize16KB) 測定結果

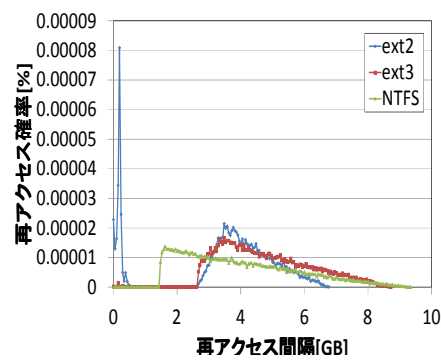


図 6. FFSB (OPsize64KB) 測定結果

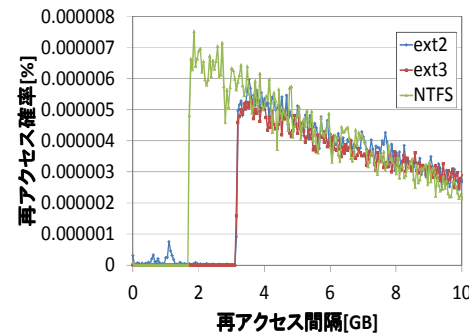


図 7. FFSB (OPsize256KB) 測定結果

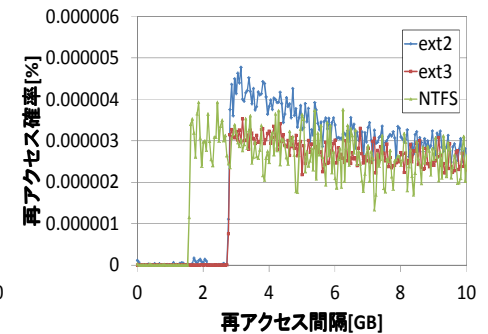


図 8. FFSB (OPsize1MB) 測定結果

5. キャッシュ置換アルゴリズムのヒット率評価

本章にて, ネットワークストレージキャッシュ環境における既存アルゴリズムの有効性を確認するために, シミュレーションによる評価を行う。評価対象となる置換手法は, LRU, RANDOM, FIX, MQ である。LRU, は 3 章にて説明した手法である。FIX はキャッシュに格納されたデータを置換せず保持し続ける手法である。RANDOM は, キャッシュから無作為に選択したデータを置換対象とする置換手法である。MQ は 6 章の関連研究にて説明する。

5.1 シミュレーション環境

シミュレーション環境は以下の通りである。ネットワークストレージには 1 台のサーバー計算機が接続されており, サーバー計算機のキャッシュ置換手法とし LRU を用いていることを想定している。シミュレーションには 4 章の FFSB の測定で得た SCSI 層のアクセスログを使用した。

表 3.シミュレーションパラメータ

ストレージキャッシュ	1,000,000[ブロック]
サーバーキャッシュサイズ	7,256,146[ブロック]
ストレージキャッシュサイズ(等倍)	7,256,146[ブロック]
ストレージキャッシュサイズ(0.5倍)	3,628,073[ブロック]
ストレージキャッシュサイズ(1.5倍)	10,884,219[ブロック]
MQ:Qoutサイズ	ストレージキャッシュサイズ×4
使用アクセスログ	FFSB(Datasize16GB,OPsize16KB~1MB)

5.2 シミュレーション結果

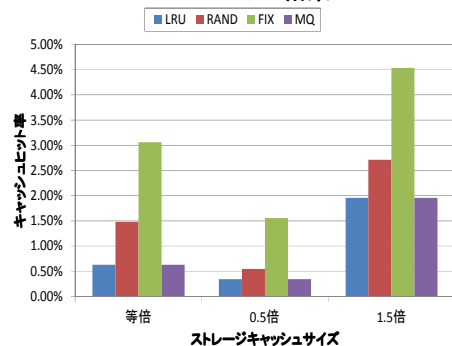


図 10. シミュレーション結果 (FFSB:0Psize16KB, FS:ext2)

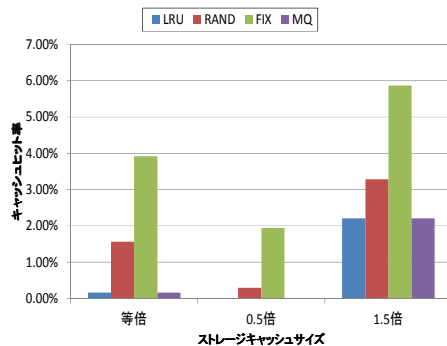


図 11. シミュレーション結果 (FFSB:0Psize16KB, FS:ext3)

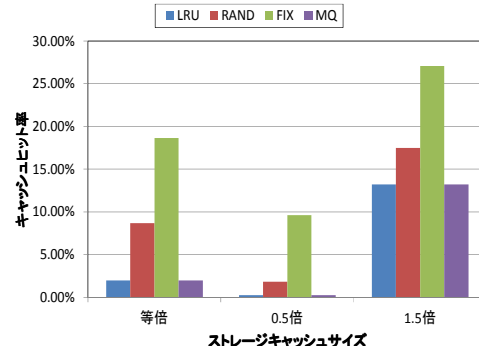


図 16. シミュレーション結果 (FFSB:0Psize256KB, FS:ext2)

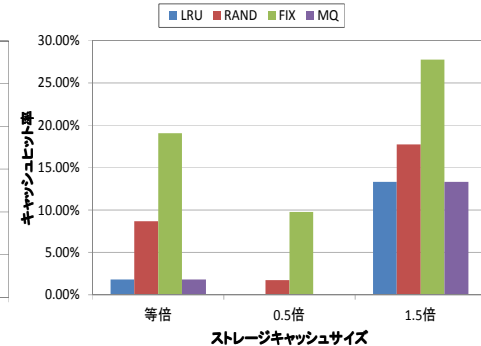


図 17. シミュレーション結果 (FFSB:0Psize256KB, FS:ext3)

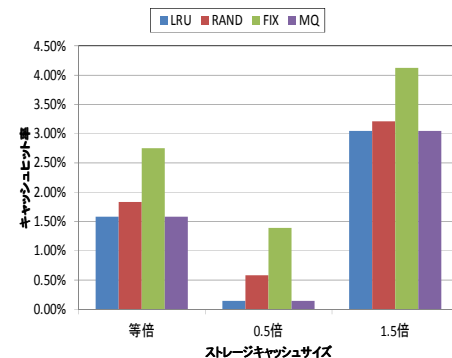


図 12. シミュレーション結果 (FFSB:0Psize16KB, FS:NTFS)

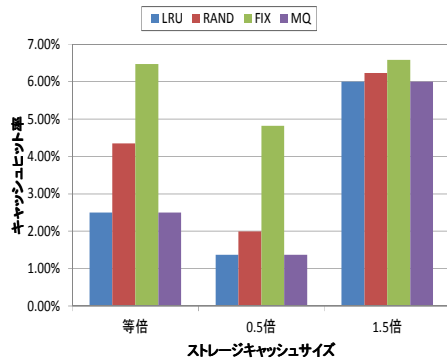


図 13. シミュレーション結果 (FFSB:0Psize64KB, FS:ext2)

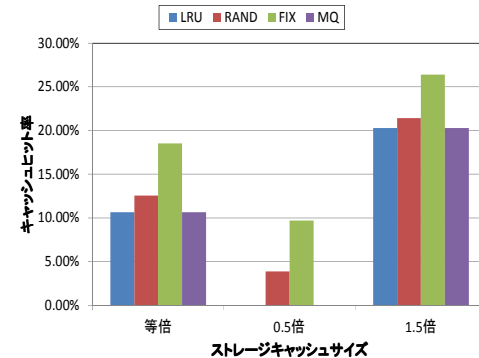


図 18. シミュレーション結果 (FFSB:0Psize256KB, FS:NTFS)

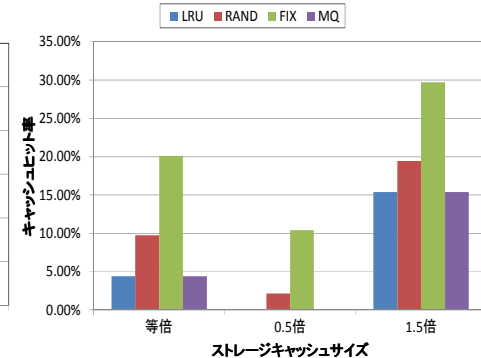


図 19. シミュレーション結果 (FFSB:0Psize1MB, FS:ext2)

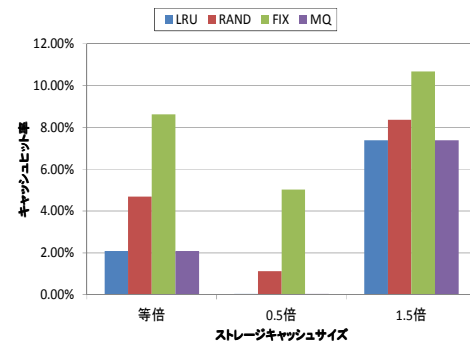


図 14. シミュレーション結果 (FFSB:0Psize64KB, FS:ext3)

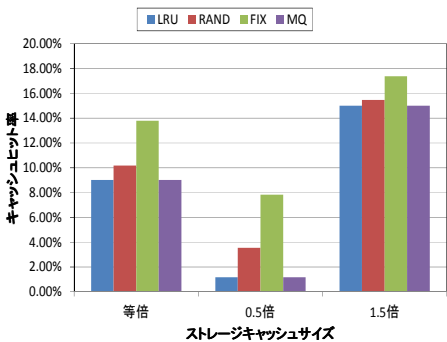


図 15. シミュレーション結果 (FFSB:0Psize64KB, FS:NTFS)

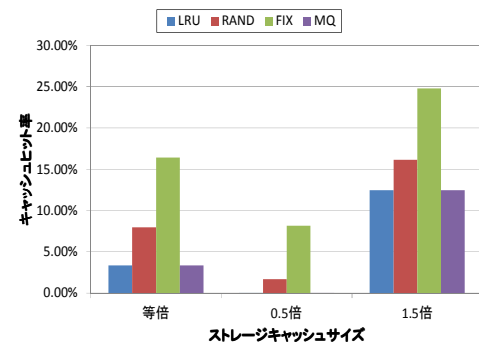


図 20. シミュレーション結果 (FFSB:0Psize1MB, FS:ext3)

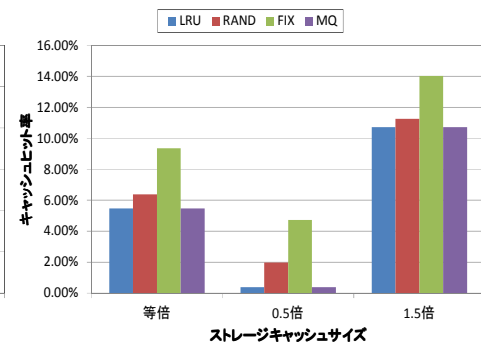


図 21. シミュレーション結果 (FFSB:0Psize1MB, FS:NTFS)

キャッシュヒット率の評価結果を図 10~21 に示す。横軸は、サーバー計算機キャッシュサイズとストレージキャッシュの比を表しており、“等倍”は、サーバー計算機キャッシュとストレージキャッシュのサイズが等しく、“0.5 倍”と“1.5 倍”は、ネットワークストレージキャッシュのサイズがサーバー計算機キャッシュの 0.5 倍と 1.5 倍である。

図 10~図 21 の全ての場合において LRU および MQ のヒット率が極めて低く、ネットワークストレージキャッシュにおいてこれらの手法が効果的に機能しないことがわかった。RANDOM は LRU や MQ よりは性能が高いが FIX よりも低く、これも効果的な手法とは考えづらいと言える。

サーバー計算機キャッシュのサイズと、ネットワークストレージキャッシュのサイズの比率について注目すると、ネットワークストレージキャッシュのサイズが小さいほど LRU と MQ の性能が悪くなっていることがわかる。これは、ネットワークストレージキャッシュのサイズがサーバー計算機キャッシュのサイズと比較して相対的に小さくなるほど、ネットワークストレージキャッシュのデータがサーバー計算機キャッシュに包含される確率が増え、負の参照の局所性が強まっているからだと言える。

次にファイルシステムの違いに着目すると、前章にて最も明確に負の参照の局所性が現れた ext3 において LRU と MQ のヒット率が低くなっていることがわかる。ext3 に次いで NTFS における LRU と MQ の性能が悪く、負の参照の局所性が最も弱かった ext2 においては LRU と MQ の性能劣化が一番小さくなっている。

最後にオペレーションサイズの違いに着目すると、オペレーションサイズに依らず LRU と MQ のキャッシュヒット率が低いことがわかる。

MQ は、下位キャッシュ用に提案されたキャッシュ置換アルゴリズムであるが、今回の評価では LRU とほぼ同等の性能となり、非常に低い性能となった。MQ は参照回数によって、格納データをどの LRU キューに保存するか決定し、標準的決定方法として $\log_2(\text{参照回数})$ により決定する方法が示されている[2]。キュー数を標準の 8 本[2]にした場合、参照回数が 128 回以上になってしまうと、全てのデータが最上位の LRU キューに格納されてしまい、実質的に MQ の動作が LRU と類似の動作になったことが大きな原因と考えられる。このことから、MQ を効果的に使用するにはパラメータを環境毎に適切に変更する必要がある、最適化を行えないと性能が極めて悪くなってしまうことがわかる。

6. 関連研究

6.1 キャッシュに関する研究

近年では、プロセッサの動作速度とメモリの動作速度、HDD の動作速度の差が激しくなっている。このため計算量や記憶容量だけを指標にしたのではアルゴリズム

の優越を判断するのは難しい。この現状を考慮して、メモリ参照の局所性の指標を定義することを目的とし、時間的・空間的近傍を用いてヒット・ミス率を確率として定義する研究が行われている[4][5]。

6.2 MQ(Multi-Queue)

ネットワークストレージのような下位キャッシュのための置換手法として MQ(Multi Queue)が提案されている[2]。下位キャッシュには負の参照の時間的局所性が存在しており、近い将来再度参照されることはない。このため、MQ は長時間データを保持することにより、キャッシュヒット率を高めることを目的としたアルゴリズムである。MQ は複数の LRU キューにより構成されており、各データは参照回数によって格納されるキューが選ばれる。格納されたデータには expire time が設定されており、あまりに長期間アクセスが無かったデータは、下位のキャッシュに移される。置換対象となるデータは最も下位のキューの中で最も古いデータが選ばれ、置換対象となったデータの ID およびその参照回数は Qout と呼ばれるキューに保管される。該当データが再度参照され、キャッシュに格納される際に該当データの参照回数として Qout に保管してあった値が加算され、保管キューを選択する。

6.3 ネットワークストレージを用いる二重キャッシュ環境に関する研究

文献[6]は、ネットワークストレージを用いる二重キャッシュ環境において発生するデータの重複、およびキャッシュヒット率に関する調査から、負の参照の時間的局所性について考察し、一般的な LRU 置換アルゴリズムは有効ではないことを示している。また LRU を用いず、キャッシュ内容を固定化することで性能が向上することを示した。

ネットワークストレージ環境における既存のキャッシュ置換手法の評価として、Wilick らによる性能評価がある[7]。彼らはシミュレーションにより性能評価を行い、LRU がネットワークストレージ環境において高い性能を示すことができず、LFU の方が高い性能を示すことを確認している。

また Luo らは、ネットワークストレージのフロントエンドとしてプロキシを配置してキャッシュヒット率ではなくデータアクセスレイテンシを削減させる proxy cache scheme を提案している[8]。

6.4 負の参照の時間的局所性に関する研究

WC(Weighted Count)は負の参照の時間的局所性を考慮した手法である[9]。WC では LFU 同様アクセス履歴を保管することで、過去の参照情報を利用し、最も価値のないデータを選択する。前章で示したように一度参照してからの経過時間が短いデータは、サーバー計算機キャッシュに存在するため、ネットワークストレージに参照要求が届くことはない。その後約サーバーキャッシュサイズ分のアクセスを行うと、該当データはサーバーキャッシュから破棄され、ネットワークストレージ側の参照要求の到達確

率が急増する．このような負の参照の時間的局所性を考慮し，一度参照してからの経過時間によりアクセス毎の価値を決定する．

7. おわりに

本稿では，下位キャッシュへの参照の局所性の調査環境の構築を行い，下位キャッシュにおける参照の時間的局所性の解析を行い，負の参照の時間的局所性の存在の調査を行った．その結果，ファイルシステムに ext3, NTFS を用いた場合は負の参照の時間的局所性が存在し，ext2 を用いた場合は必ずしも存在しないことを確認した．また，既存アルゴリズムの有効性を確認するために，本調査で得たアクセスログを使用しシミュレーションによる評価を行った．その結果全てのシミュレーションで LRU と MQ のヒット率が低く，多段キャッシュ環境において，LRU が効果的に機能しておらず，また MQ も標準的な使用方法では性能が低いことが分かった．今後は負の参照の時間的局所性を考慮したキャッシュ置換アルゴリズムの提案を行っていく予定である．

謝辞

本研究は科研費（22700039）の助成を受けたものである．

参考文献

- [1] 長廻雄介，山口実靖，“多段キャッシュ型ネットワークストレージへのアクセスの時間的局所性を考慮したキャッシュ置換法” 情報処理学会 2010 年全国大会 5ZA-5.
- [2] Yuanyuan Zhou and James F. Philbin, Kai Li, “The Multi-Queue Replacement Algorithm for Second Level Buffer Caches,” In Proceedings of the 2001 USENIX Annual Technical Conference, 2001.
- [3] P.J. Denning, “The Locality Principle,” Communications of the ACM, Volume 48, Issue 7, pp. 19–24, (2005)
- [4] 田中淳裕，“キャッシュのモデル化とその応用”，社団法人日本オペレーションズ・リサーチ学会，pp.434-437, 20040701
- [5] 田中淳裕，“メモリ参照の局所性に着目したソフトウェア性能評価手法”，日本ソフトウェア科学会，pp.120-134, 19990315
- [6] 宮野新平，山口実靖，浅谷耕一，“多段キャッシュ型ネットワークストレージへのアクセスの時間的局所性を考慮したメモリキャッシュ制御”，社団法人情報処理学会，pp.7-12, 20090226
- [7] D. L. Willick, D. L. Eager, and R. B. Bunt, “Disk Cache Replacement Policies for Network Fileservers,” In ICDCS, May 1993
- [8] Yihui Luo and Changsheng Xie, Xinwei Zheng, Chengfeng Zhang, Zhen Zhao “A Proxy Cache Scheme of Network Storage” IEEE Embedded Software and Systems, 2005
- [9] Yusuke Nagasako and Saneyasu Yamaguchi, “A Server Cache Size Aware Cache Replacement Algorithm for Block Level Network Storage,” The Fourth International Workshop on Ad Hoc, Sensor and P2P Networks (AHSP2011), Jun. 2011