

## 二重キャッシュ環境における負の参照の時間的局所性を考慮したキャッシュ置換

杉本 洋輝<sup>†</sup> 山口 実靖<sup>†</sup>

工学院大学 工学部 情報通信工学科<sup>†</sup>

### 1. はじめに

仮想化環境下において、ホスト OS キャッシュ(下位キャッシュ)へのアクセスはゲスト OS キャッシュ(上位キャッシュ)を介して行われる。このような二重キャッシュ環境下において、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ことが実 OS で確認されている[1]。また、二重キャッシュ環境ではゲスト OS キャッシュとホスト OS キャッシュに同一のデータを重複して格納している可能性が高い。

ホスト OS キャッシュは負の参照の時間的局所性やゲスト OS キャッシュとのデータの重複により有効に機能しない。そのため、負の参照の時間的局所性とデータの重複を考慮した置換手法が必要である。

### 2. 負の参照の時間的局所性

アプリケーションが発行するデータアクセス要求には多くの場合偏りがあり、LRU など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。多くの場合、参照の時間的局所性を考慮したメモリ置換手法を用いることにより下層の低速記憶装置の記憶容量よりも小さいキャッシュでも十分な性能を発揮することができる。

しかし、仮想化環境のような二重キャッシュ環境では通常とは逆向きの負の参照の時間的局所性が存在する。ホスト OS キャッシュへのアクセスは、図 1 のようにゲスト OS キャッシュを介して行われる。アプリケーションから発行された要求がゲスト OS キャッシュ、ホスト OS キャッシュの両キャッシュでミスヒットした場合、両キャッシュには同一のデータが格納される。しかし、最近アクセスされたデータへのアクセス要求はゲスト OS キャッシュ上で処理され、ホスト OS キャッシュに届くことはない。従ってホスト OS キャッシュでは“最近アクセスされたデータは近い将来再度アクセスされる可能性が低い”という通常とは逆向きの負の参照の時間的局所性が存在する。

また、負の参照の時間的局所性はベンチマークのデータサイズが十分に大きい時、上位キャッシュのサイズが大きいくほど影響が大きくなることが確認されている[1]。

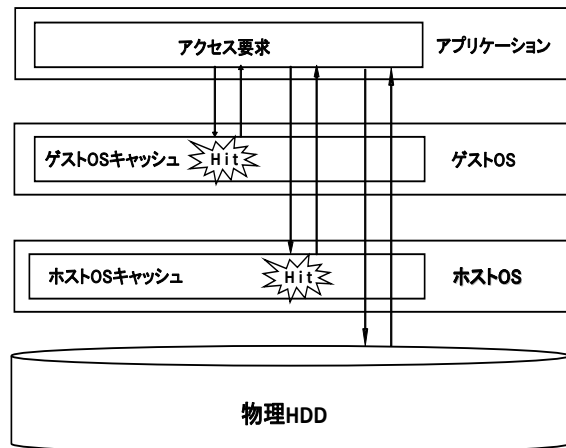


図 1 二重キャッシュ構造

### 3. 負の参照の時間的局所性を考慮したキャッシュ置換

#### 3.1 負の参照の時間的局所性を考慮したキャッシュ置換

上位キャッシュ、下位キャッシュ両キャッシュでミスヒットした場合、両キャッシュには HDD から読み込まれた同じデータが格納される。しかし、最近参照されたデータへの要求は上位キャッシュで処理され下位キャッシュには届かない。これを考慮し、ホスト OS キャッシュでは最後に参照されてからの時間が最も短いデータを破棄対象とする (MRU) 手法を提案する。また、上位キャッシュから破棄されるデータを監視し、破棄データを下位キャッシュに最後に参照されてからの時間を最長 (最も破棄されにくい状態) として格納する。上位キャッシュから破棄されたデータを格納する理由は、上位キャッシュから破棄されたデータは上位キャッシュに存在しないため、両キャッシュでのデータの重複が起きないためである。

この手法により上位キャッシュと下位キャッシュのデータの重複を抑えることができ、キャッシュヒット率の向上が期待できる。

#### 3.2 負の参照の時間的局所性を考慮したキャッシュ置換の評価

シミュレーションによりキャッシュヒット率を評価した。シミュレーションではキャッシュおよびディスクは 4KB ブロックに管理されているものとした。上位キャッシュの置換アルゴリズムには LRU を用い、下位キャッシュの置換アルゴリズムを変えてシミュレーションを行った。

A Cache Replacement Algorithm in Virtualized Environment with Two Caches

<sup>†</sup>Hiroki Sugimoto, Saneyasu Yamaguchi

<sup>†</sup>Department of Information and Communications Engineering, Kogakuin University

図2～図5にシミュレーション結果を示す。

まず、LRUのヒット率に着目して考察する。シミュレーションの結果ではLRUは今回使用したアルゴリズムの中で最もヒット率が低かった。上位キャッシュのサイズを増やすと下位キャッシュのサイズが同じ場合でもヒット率が低下する事がわかる。この理由は2つ考えられ、一つは上位キャッシュのサイズを大きくしたことにより負の参照の時間的局所性の影響が大きくなったため、参照の時間的局所性を期待しているLRUが効果的に機能しなかったためと考えられる。もう一つの理由として、上位キャッシュのサイズが大きくなったことにより、両キャッシュのデータの重複が増えたため、上位キャッシュでミスヒットしたデータは下位キャッシュでもミスヒットする可能性が高くなり、ホストOSキャッシュヒット率の低下に繋がったと考えられる。

次に、下位キャッシュの置換アルゴリズムに提案手法を選択した場合のヒット率に着目して考察する。下位キャッシュの置換アルゴリズムに提案手法を選択した場合は今回使用したアルゴリズムの中で最もヒット率が高かった。また、LRUのヒット率が上位キャッシュの増加とともに低下するのに対し、提案手法のヒット率は上位キャッシュの増加とともに向上する。上位キャッシュの増加とともにヒット率が向上した理由として、提案手法では負の参照の時間的局所性を考慮し、最近参照のあったデータを破棄対象とするためと考えられる。また、提案手法では上位キャッシ

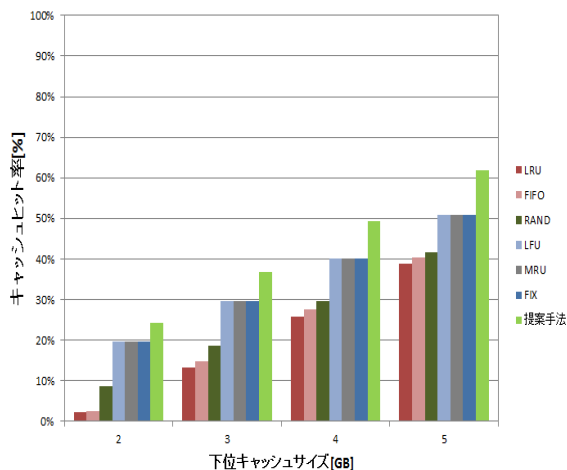


図2 シミュレーション結果, 上位キャッシュ 2GB

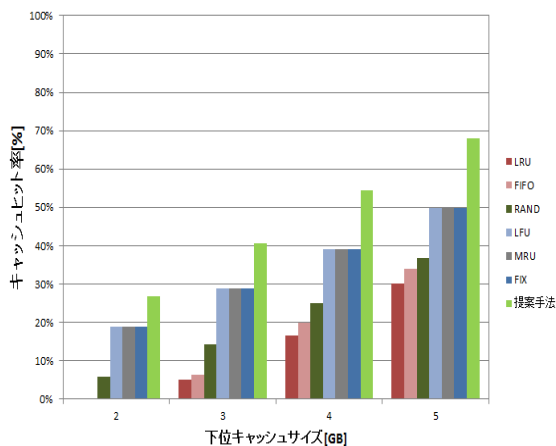


図3 シミュレーション結果, 上位キャッシュ 3GB

ュから破棄されるデータを下位キャッシュに格納することにより両キャッシュのデータの重複を抑えることができる。そのため、上位キャッシュでミスヒットしたデータでも下位キャッシュに当該データがある可能性が高いためヒット率が向上したと考えられる。

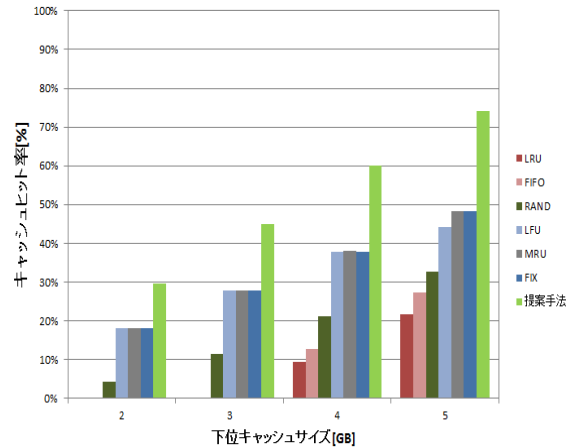


図4 シミュレーション結果, 上位キャッシュ 4GB

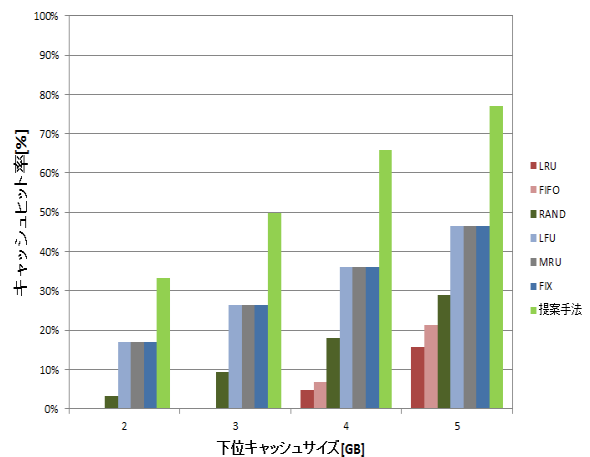


図5 シミュレーション結果, 上位キャッシュ 5GB

#### 4. おわりに

本稿では、仮想化環境下における負の参照の時間的局所性の紹介と負の参照の時間的局所性を考慮した置換手法の提案をし、評価した。その結果、提案手法は多くのOSで採用されている置換アルゴリズムであるLRUよりも高い性能を示すことがわかった。

今後は、代表的なホストOSであるLinuxへの実装を行っていく予定である。

#### 謝辞

本研究は JSPS 科研費 24300034, 25280022 の助成を受けたものである

#### 参考文献

[1] 竹内洗祐, 山口実靖, “複数サーバ接続ネットワークストレージ環境での参照の局所性の解析”第24回 コンピュータシステム・シンポジウム (ComSys 2012)