

ウェイ適応型キャッシュのための低消費エネルギー指向挿入ポリシー

東方 雄亮[†] 佐藤 雅之[†] 江川 隆輔^{††,†††}
滝沢 寛之^{†,†††} 小林 広明^{††,†††}

1. はじめに

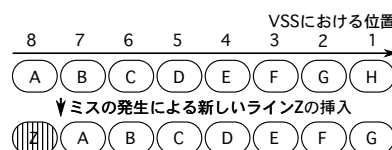
近年、キャッシュメモリ (以下、キャッシュ) の低消費エネルギー化が求められている。キャッシュの消費エネルギーを削減するために、ウェイ適応型キャッシュが提案されている¹⁾。ウェイ適応型キャッシュは、性能の維持に必要なキャッシュウェイの電源供給を停止することによって、キャッシュの低消費エネルギー化を実現する。しかし、キャッシュのライン管理に LRU 置換ポリシーを用いているため、キャッシュへ配置された後に一度も再利用されないライン (以下、再利用性のないライン) がキャッシュを占有する可能性がある。性能向上には貢献しない再利用性のないラインが占有するウェイへ電力を供給することは、エネルギー効率を著しく低下させる。

本研究では、再利用性のないラインがキャッシュに長期間保持されることを防ぐため、ウェイ適応型キャッシュのための Victim Selection Stack- K 挿入ポリシーを提案する。提案ポリシーに基づいて再利用性のないラインをキャッシュから早期に追い出すことによって、再利用性のないラインのキャッシュ占有を防ぎ、性能低下なく電源供給を停止できるウェイの数が増加する。このため、ウェイ適応型キャッシュの消費エネルギーのさらなる削減が期待できる。

2. 既存の挿入ポリシーの問題点

ウェイ適応型キャッシュでは、置換されるキャッシュラインの選択に LRU 置換ポリシーが用いられている¹⁾。LRU 置換ポリシーは一般的に広くキャッシュに用いられているが、再利用性のないラインが長時間維持されるデータ管理ポリシーである。

本研究では、キャッシュセット毎に下位のメモリ階層へライトバックされるラインを選択するための順序列である Victim Selection Stack (VSS) を考える。LRU 置換ポリシーによるライン置換における VSS を図 1 に示す。VSS において最も低い位置 (VSS-1) に存在するラインは次回のライン置換においてライトバックさ



れる。ウェイ適応型キャッシュ機構で N 個のウェイが有効化されている場合、LRU 置換ポリシーでは、新しいラインは VSS- N の位置に挿入される。VSS- N にあるラインが置換されるためには、ラインが VSS-1 の位置に到達する必要がある。その結果、再利用されるラインを保持するために必要なウェイ数が増え、キャッシュの消費エネルギーは増大する。したがって、再利用性のないラインの占有を低減することによって、さらなる消費エネルギーの削減が期待できる。

再利用性のないラインを早期に追い出すポリシーとして、LRU 挿入ポリシー²⁾が提案されている。LRU 挿入ポリシーによるライン置換における VSS の変化を図 2 に示す。LRU 挿入ポリシーでは、すべての新たなラインは VSS-1 の位置に挿入される。再利用性のないラインが挿入された場合、VSS-1 に挿入された後にアクセスされることはない。そのため、次のミス発生時のライン置換で再利用性のないラインを即座にキャッシュから追い出すことができる。

このように、LRU 挿入ポリシーでは LRU 置換ポリシーより早期に再利用性のないラインを追い出すことができ、再利用性のないラインがキャッシュを占有する割合を減らすことができる。この結果、ウェイ適応型キャッシュにおいて電源供給されるウェイ数を削減することが可能である。しかし、LRU 挿入ポリシーでは再利用されるライン (以下、再利用性のあるライン) であっても、常に VSS-1 の位置に挿入されるため、このようなラインの多くが再利用される前に追い出され、キャッシュヒット率が低下する恐れがある。

[†] 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
^{††} 東北大学サイバーサイエンスセンター
Cyberscience Center, Tohoku University
^{†††} JST CREST



図3 VSS- K 挿入ポリシーを用いたキャッシュセット ($K = 3$)

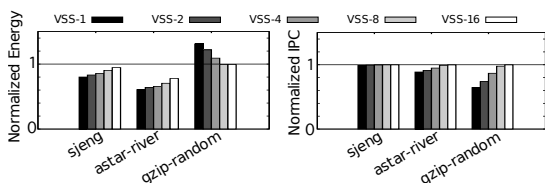


図4 LRU 置換ポリシーを用いた場合によって正規化される VSS- K 挿入ポリシーを用いたウェイト適応型キャッシュの消費エネルギーと IPC

3. VSS- K 挿入ポリシーの提案

本研究では、再利用性のあるラインが再利用される前に追い出されることを抑制しつつ、再利用性のないラインを早期に追い出すことを目的に、VSS- K 挿入ポリシー ($1 \leq K \leq N$) を提案する。VSS- K 挿入ポリシーによるライン置換における VSS の変化を図 3 に示す。VSS- K 挿入ポリシーは、キャッシュミスが発生した場合、新しいラインを VSS の K 番目に高い位置に挿入する。また、キャッシュヒットが発生した場合、アクセスされたラインが VSS- N に再配置される。VSS の低い位置へ新たなラインを挿入することによって、LRU 置換ポリシーと比較して再利用性のないラインの追い出しが早くなると同時に、参照の局所性と VSS におけるラインの再配置によって、LRU 挿入ポリシーよりも再利用性のあるラインを長くキャッシュ中に維持することができる。このため、性能低下を抑制しつつ電源供給されるウェイト数のさらなる削減を期待できる。なお、VSS-1 挿入ポリシーは LRU 挿入ポリシーと、VSS- N 挿入ポリシーは LRU 置換ポリシーと同等である。したがって、VSS- K 挿入ポリシーは LRU 置換ポリシーや LRU 挿入ポリシーをより一般化したポリシーと考えることができる。

4. 性能評価

本研究では、SPEC CPU ベンチマークスイート³⁾にある *sjeng*, *astar-river*, *gzip-random* という 3 つのアプリケーションを用いて提案手法の有効性を評価する。ここで *sjeng* は、キャッシュに置かれたラインが次にアクセスされるまでの時間が比較的短いアプリケーションである。一方、*gzip-random* はその期間が長いアプリケーションである。*astar-river* は両者の中間的な特徴を持つアプリケーションである。

VSS- K 挿入ポリシーを用いたウェイト適応型キャッシュの消費エネルギーと性能を図 4 に示す。まず *sjeng* では、 K が小さい場合でも、各ラインがキャッシュから

追い出される前に再利用される。また、再利用性のないラインはキャッシュから早期に追い出されるため、より多くのウェイトへの電源供給を停止できる。その結果、 K を小さくすることによって、*sjeng* ではほとんど性能を低下させずに消費エネルギーを削減できることがわかる。また、*astar-river* では一部のラインが再利用される前にキャッシュから追い出されるため、 K の減少にしたがって性能が低下している。しかし、 K を小さくすることによって、より多くのウェイトへの電源供給を停止できるため、性能低下にも関わらず消費エネルギーを削減できている。この結果から、*sjeng* や *astar-river* のように再利用までの期間が比較的短いアプリケーションにおいて、提案ポリシーはキャッシュの低消費エネルギー化に有効であることがわかる。

しかし、*gzip-random* では、 K を小さくすることによって性能が大きく低下するため、結果として消費エネルギーが増加している。これは、 K を小さくすることによって、再利用性のあるラインがキャッシュから早期に追い出され、キャッシュヒット率が低下するためである。この結果から、各ラインの再利用されるまでの期間が長い *gzip-random* のようなアプリケーションに対しては、提案ポリシーのパラメータである K を適切に設定する必要があることが明らかになった。このため、再利用される前にラインができるだけ追い出されない挿入位置を予測し、 K を自動選択する方法の確立が今後の課題である。

5. おわりに

ウェイト適応型キャッシュの消費エネルギーの削減を目的として、本研究では VSS- K 挿入ポリシーを提案した。評価結果より、適切な VSS への挿入位置を選択することによって、性能の低下が小さいウェイト適応型キャッシュのさらなる消費エネルギーの削減が可能であることが明らかになった。したがって、今後の課題は適切な VSS への挿入位置を自動的に選択する方法を確立することである。

謝辞 本研究の一部は日本学術振興会・戦略的創造研究推進事業によるものである。

参考文献

- 1) Kobayashi, H., Kotera, I. and Takizawa, H.: Locality Analysis to Control Dynamically Way-Adaptable Caches, *ACM SIGARCH Computer Architecture News*, Vol. 33, No. 3, pp. 25–32 (2005).
- 2) Qureshi, M. K., Jaleel, A., Patt, Y. N., Steely, S. C. and Emer, J.: Adaptive Insertion Policies for High Performance Caching, *ACM SIGARCH Computer Architecture News*, Vol. 35, No. 2, pp. 381–391 (2007).
- 3) Standart Performance Evaluation Corporation: SPEC CPU2000 Benchmark Suite, SPEC CPU2006 Benchmark Suite, <http://www.spec.org>.