Dynamic Cache Partitioning based on Dead Block Prediction

Yoshimune Saito (1010026)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 12, 2015

Keywords: Multi-core processor, Shared Cache, Cache Partitioning, Dead Block.

1 Introduction

Modern processors contain multiple cores which can concurrently execute multiple applications on a single chip. It is called Multicore processor. Multicore processors usually employ a shared cache memory used by multiple cores as last-level cache. The shared cache is efficiently used by cores. however, there is a problem that increases misses by conflict between cores. To solve the problem, dynamic cache partitioning has been studied by many investigators. The partitioning divides a cache memory among cores, and changes dynamically the partition sizes by predicting to improve performance of the cache. Conventional dynamic cache partitioning predicts a core that increase cache hits by increased partition sizes. In constrast to this, this thesis proposes novel dynamic cache partitioning that predicts a core that does not increase misses by decreasing the partition sizes.

2 Related Works

Stone studied the optimal partitioning of cache memory between some conflict processes to minimize the overall miss-rate of a cache[1]. The

literature defines "Marginal Gain", $g_j(x)$, which is the amount of miss reduction for the $thread_j$ during the period when the number of allocated cache blocks increases from x to $x + 1$. This method change each partition size depend "Marginal Gain" on the period.

This partitioning is based on off-line profiling. Suh, et al. proposed to predict optimal partition sizes based on the past pseudo "Marginal Gain"[2]. This mechanism assumes that the cache uses the standard LRU replacement policy and $g_j(x)$ represents the number of hits on x most recently used cache blocks of the $thread_j$ in the previous period.

This method has many hardware-counters. Ogawa, et al. investigated cache allocation named HFCA that do not need hardware counters[3]. HFCA does not count hits or misses. They divide each cache set into private partition(PP) used by a core and shared partition used by all cores. HFCA increases a PP size when a core hits a block in SP.

3 Proposed Method

This thesis proposes a novel dynamic cache partitioning technique that predicts a core that does not increase misses even if the partition size is decreased. Our method predicts such a core based on dead block prediction. Our mechanism decreases the partition size when the dead block predictor detects multiple dead blocks in the partition. New partition size is calculated by equation (1). The blocks excluded from the shrinked partitions are distributed to other partitions with no dead blocks.

The dead block predictor regards a block accessed a long time ago as a dead block. The time threshold is dynamically calculated by equation (3).

$$partition\_size_{new} = partition\_size_{old} - (deadblocknum - 1) \quad (1)$$

$$access\_interval = current\_time_(thread) - last\_access_(block) \quad (2)$$

$$long\_time_{new} = (long\_time_{old} + access\_interval * \delta))/2 \quad (3)$$

Our method redistributes the excluded blocks equally among the other partitions with no dead blocks. This is because our method cannot predict a core that increases hits by increasing the partition size.

4 Evaluation

The proposed method is compared with the other methods in simulation. Evaluation criteria are the followings.

$$IPC_{sum} = \sum(IPC_i) \quad (4)$$

$$WeightedSpeedup = \sum(IPC_i/SingleIPC_i) \quad (5)$$

The results showed that our method is better than HFCA by 1.4%. However, the method by Suh et al. is a little better than the proposed method.

5 Conclusion

This research proposed a novel dynamic cache partitioning technique that predicts a core that does not increase misses by decreasing partition sizes. This mechanism decreases the size of a partition with multiple dead blocks.

References

[1] Harold S. Stone, Fellow, IEEE, John Turek, Member IEEE, and Joel L. WOlf "Optimanl Partitioning of Cache Memory" IEEE TRANSACTION ON COMPUTER VOL.41, NO. 9, SEPTEMBER 1992

[2] G.Edward Suh, Larry Rudolph, and Srinivas Devadas "Dynamic Partitioning For Shared Cache Memory" The Journal of Supercomputing, 2002, July

[3] 小川周吾, 入江英嗣, 平木敬"アクセス履歴の不要なマルチコアCPU向け共有キャッシュ配分方式" 先進的計算機版システムシンポジウム SACSIS 2010, 2010, May pp 267-276