CS211 Advanced Computer Architecture Paper Reading 1

Anonymous

1 Paper Information

T. Zheng, H. Zhu and M. Erez, "SIPT: Speculatively Indexed, Physically Tagged Caches," 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2018, pp. 118-130, doi: 10.1109/HPCA.2018.00020.

2 Paper Intended Solve Problem

Current processor typically use VIPT cache architecture, which is limited by the size of a virtual page size. This policy will both increase the access latency and cache energy comsumption.
Another policy is VIVT cache architecture, which presents significant complications for cache management and coherance.
The author need to find a suitable policy that with the advantage of VIPT, includes simpism and reliability, but without the disadvantage of suboptimal L1 cache parameters(latency, capacity, and associativity).

3 Motivation

- While both capacity and associativity affect latency, associativity has the greater impact.
- At some cores, latency has a significant impact and configuration with shortest lantency performs best. But in others, cores with balanced lantency and capacity achieves the best performance.

4 Solution

4.1 Speculatively Indexed Physically Tagged

4.1.1 Method

The simplest variant of SIPT, which assuming that all necessary index bits will remain the same after translation, including those beyond the page granularity. If all speculated bits indeed are the same, then the fast access complete. If not, the cache request must be repeated with the correct index from PA, which is quite slow.

4.1.2 Accuracy

Normolly, with reasonable configuration of L1 cache, processor only require 1-3 index bits beyond the page granularity, which result in a more higher correct prediction rate. If only a single speculative index is needed, in most cases the processor can have majority of fast accessed.

4.1.3 Performance & Disadvantage

Naive SIPT has a high misspeculation rate, which result in significant gaps on total cache energy with ideal situation.

4.2 Misspeculation Prediction

4.2.1 Method

A small PC-based Perceptron predictor which make a speculate/no-speculate binary decision early in the pipline to hide cache visit lantency. The predictor has a global history register $x_1x_2\cdots x_h$ that tracks the last h speculation outcomes as ones and zeros, it also has 64 entries each being a perceptron of $h+1$ weights $w_0w_1\cdots w_h$. Perceptron calculates a prediction by performing a dot product of the history and the weights of a specific entry in the table plus a learned bias: $y = w_0 + [x_1x_2\cdots x_h]\cdot[w_1\cdots w_h]$.
If y is positive, we predict the index will not change and will continue with a fast access using the speculative index. If y is negative we bypass speculation and wait for the physical address before accessing the cache.
If the speculated bits are unchanged bt translation but the predictor chooses bypass, an opportunity for fast access was squandered. If the speculated bits are changed by translation and the predictor chooses to speculate, an extra access is generated.

4.2.2 Accuracy

The perceptron predictor achieves more than 90% accuracy in most situations, also these parameters of the perception do not show strong sensitivity to experiments.

4.2.3 Performance

The predictor practically eliminates extra accesses due to SIPT and thus saves significant energy.

4.2.4 Advantage

- Only use PC, the prediction can be overlapped with other pipline stages.
- The predictor introduces no extra lantency and only negligible area and energy overheads.

4.2.5 Disadvantage

- Fails to reduce the extra latency from slow accesses and hence cannot improve performance.

4.3 Partial Index Prediction

4.3.1 Method

In the context of SIPT, predicting values of multiple speculative tag bits is doable because of spatial locality in memory address mapping. As linux manages free pages use the buddy algorithm, it usually breaks large memory groups to satisfy bursts of memory allocation requests, which will lead to a significant amount of contiguous physical pages being mapped to a contiguous virtual address space. It means that for a range of contiguous address, the delta between a virtual address and its corresponding physical address is the same. Partial Index Prediction use index delta buffer(IDB) to predict these delta between VA to PA, it is a PC-indexed table with each entry storing a index delta. It additionally update the expected delta, which remains stable as long as the same regions are accessed.
First, queried the perception predictor, if it predicts to speculate, the speculative index is used, if it predicts to bypass speculation, the IDB is queried and its predicted index bit values are used to access the cache with a speculative index.
The first case is correctly-speculated fast access by the bypass predictor in which case the IDB is not accessed.
The second case is bypass-prediction and for which the IDB predicted the correct speculative index bits. The remaining accesses are slow and generate extra L1 accesses.

4.3.2 Accuracy

When only a single bit value needs to be predicted, over 90% of all accesses are fast accesses. With 2 and 3 speculative index bits, the combined predictor successfully convert many slow accesses into fast ones.

4.3.3 Performance

In single core, the trend that SIPT with combined speculation bypass and IDB prediction yield speedup and energy saving very close to the ideal cache configuration.
While in multicore, the average IPC improvement is 8.1%, which is slightly better than with a single core.

5 Further Discussion

5.1 Way Prediction

Way prediction and SIPT are complementary, when applied to the baseline cache, achieves 89% accuracy and reduces cache energy by 24% on average. As associativity reduced, the way prediction accuracy increases to 97.3% on average, and there is only a 0.3% performance drop compared to SIPT alone. Also it saves 2.2% additional cache energy.

5.2 Predictability of Partial Index Bits

Running with fragmented physical memory or disabling THP does degrade the behavior of SIPT, but not significant. SIPT benefits from contiguous memory mapping but does not solely rely on it. Using extremely fragmented memmory or disabling THP has limited impact.

5.3 Implications for Instruction Schedulers

SIPT is very accurate and can use existing speculative-scheduling approaches to recover from rare mis-prediction.

6 Personal Thinking

6.1 Understanding on why using PC instead of VA in perception

While reading the perception part, it is common to have doubts that why choose PC as the input, since the same PC can read/write different address, although choosing PC means the perception stage can running with the piplines at the same time. My personal understanding is, the same PC reading different address often means a situation, where the PC is in a loop and the addresses are likely to be in the same page. As the page is 4KiB or even larger, the speculated bits of same PC of different VA are likely to be the same.

6.2 Apply multi perception on speculation

Since the current solution of this article is, the perception is designed to decide whether it should speculate or not, the IDB is designed to caculate what actually value is of the speculated bits.
As the IDB may cause more overhead, additonal latency and energy ineffiency, I want to apply multi perception on speculation. Instead of using IDB to get the actual value, I directly use more perceptions(or multi-label perception) to speculate bits.
The shortage of this try might be as follows, first, due to the high dimension of the perception, it will take a more long time to get a stable weights with high accuracy; second, it's stabilization might not as well as the Partial Index Prediction, since different VAs with same gap can perform different PAs' features on the multi-perception.
The advantage is obvious since this multi-perception implementation can be realized more easily on hardware, more energy effient, more shorter lantency and more less overhead.