Rank Idle Time Prediction Driven Last-Level Cache Writeback

Zhe Wang, Samira M. Khan, Daniel A. Jiménez
University of Texas at San Antonio

Motivation and Objective

Motivation

- Memory access latency is a major performance bottleneck.
- Write-induced interference [1] delays subsequent read requests for hundreds of cycles in a modern DDRx memory system. Without write-induced interference, read latency can decrease 25% on average in a 4-core processor with DDR3 memory system as shown in figure 1.
- Ranks are idle 38% on average in a 4-core processor with DDR3 memory configuration as shown in figure 2.

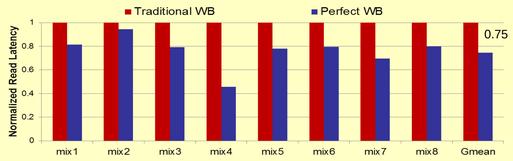

Figure 1 read latency by using traditional writeback and perfect writeback

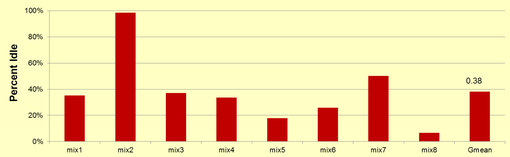

Figure 2 Rank idle percentage

Contribution

Develop a prediction driven LLC writeback technique. This technique uses a rank idle predictor to predict when a rank will have significant idle time. A sequence of scheduled dirty cache blocks can be written back during this idle rank period. Write-induced interference is significantly reduced by our technique.

Methodology

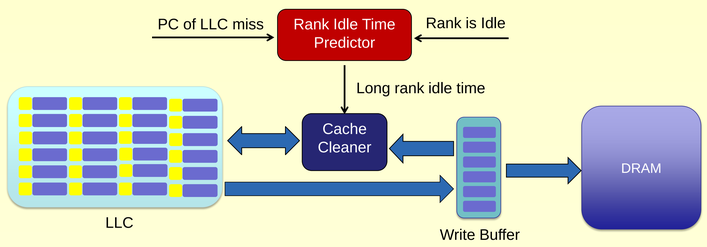

Figure3 System structure

System Structure

Figure 3 illustrates the structure of our technique. A two-level predictor is used to predict long stretches of idle rank cycles for a given rank. A sequence of scheduled dirty cache blocks that are generated by the Cache Cleaner [2] are written back during a predicted long idle period.

Rank Idle Predictor

Figure 4 shows the structure of the two-level predictor. Two levels are used so that if the first predictor mispredicts a long idle period, the second predictor has another chance to predict this long idle period. The prediction state consists of a table of two bit saturating counters, much like a branch predictor. The predictor table is indexed by the address (PC) of the instruction. The rank idle cycle counter is used to count the number of idle cycles.

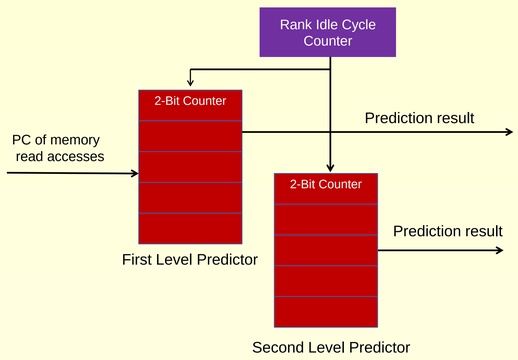

Figure 4 A two-level rank idle predictor

```
Function PredDrivenSched
Begin
        if rank_idle_cycles==1 then
            prediction = first_predictor_predict
        end
        else if rank_idle_cycles == n
            prediction = second_predictor_predict
        end
        else if rank_idle_cycles == k
            prediction = true
        end
        if prediction == true then
            call schedule_writeback
        end
end
```

Figure 5 Prediction driven LLC writeback scheduling algorithm

Rank Idle Time Prediction Driven Writeback Scheduling

A sequence of s scheduled dirty cache blocks will be written back to DRAM during the predicted rank idle cycles. Figure 5 shows the prediction driven LLC writeback scheduling algorithm. We incorporate the rank idle prediction with the parallelism-aware writeback technique; that is write back a sequence of dirty cache blocks from the LLC that preserve the bank-level parallelism in a particular rank.

Evaluation and Conclusion

Methodology

Execution core	8core CMP, out of order, 256 entry reorder buffer
Caches	L1 I-cache: 32KB/2 way, private, 64 bytes block size, LRU, 2-cycle L1 D-cache: 32KB/2 way, private, 64 bytes block size, LRU, 2-cycle L2 Cache: 16MB/16 way, shared, 64 bytes block size, LRU 14-cycle
Memory	2 memory controller, 2 ranks per channel, 8 banks/rank, 8K bytes row buffer per bank, DDR3-1600 11-11-11, 32 entry per channel write buffer

Table 1 System configuration

We use Marssx86 simulator together with DRAMSim2 to model the memory system. The system configuration is shown in Table 1. We use SPEC 2006 benchmarks for the evaluation. We run six groups of 8-core workloads. 8 of the 24 benchmarks are randomly chosen to run in the same pass.

Performance Evaluation

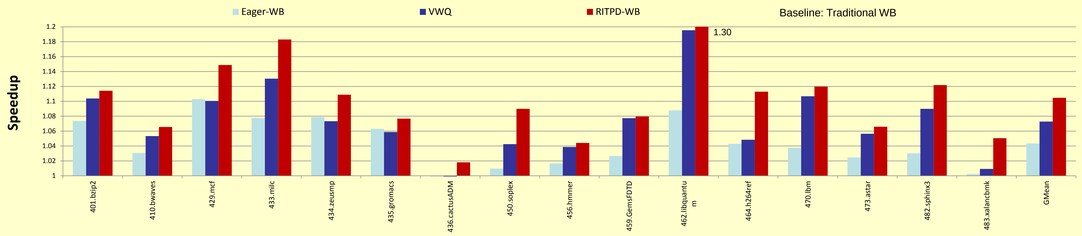

Figure 5 IPC Speedup using various optimization over baseline

Figure 5 shows the IPC speedup of various writeback schemes over traditional writeback. We can see the prediction-guided writeback schemes have better performance over other techniques. Our technique yields 10.5% speedup on average.

Prediction Evaluation

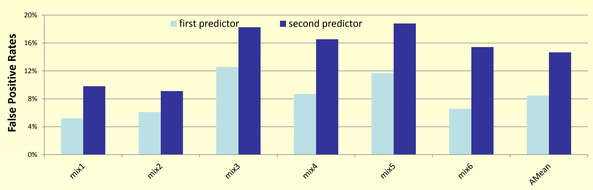

Figure 6 Experiment results for false positive rates

False Positive Rate

The false positive rate is calculated by the number of mispredicted positive predictions divided by the total number of predictions. False positives are harmful because they wrongly allow the short rank idle periods to service LLC wrtiebacks.

Figure 6 shows the false positive rates for the two-level predictor. False positive rates for the first-level and second-level predictors are 8.5% and 14.7% on average, respectively.

Read Latency Evaluation

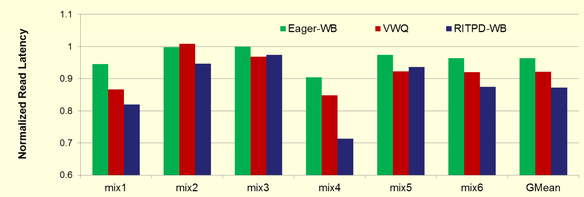

Figure 7 Read latency evaluation results

Read Latency Evaluation

Figure 7 shows the read latency normalized to traditional writeback, the RITPD-WB technique reduces the write-induced interference to read accesses, thus reducing the average read latency. The RITPD-WB technique reduces the read latency on average by 12.7%.

Conclusion

We propose a rank idle time prediction driven LLC writeback technique. This technique shows a significant performance improvement when the rank idle predictor works with LLC scheduled writebacks. In future work, we plan to investigate the use of the rank idle predictor for other optimizations to improve the memory efficiency.

References

[1] C. J. Lee, V. Narasiman, E. Ebrahimi, O. Mutlu and Y. N. Patt. Dram-aware last-level cache writeback: Reducing write-caused interference in memory system. In HPS Technical Report.
[2] Jeffrey Stuecheli, Dimitris Kaseridis, David Daly, Hillery C. Hunter, and Lizy K. John. The virtual write queue: coordinating dram and last-level cache policies. ISCA '10, pages 72–82, New York, NY, USA, 2010. ACM.
[3] Hsien-Hsin S. Lee, Gary S. Tyson, and Matthew K. Farrens. Eager writeback - a technique for improving bandwidth utilization. MICRO33, pages 11–21, New York, NY, USA, 2000. ACM.