Brief Work Report on 05/09/2011

Min Cai

May 9, 2011

1 LLC Content Management Techniques for Efficient Software Based Helper Threaded Data Prefetching on CMPs

Note: here I study two forms of cache content management[4]: (1) passive: LLC replacement; (2) active: hardware constructed p-thread mechanism, to improve the efficiency of the HT scheme.

1. Implemented inter-thread reference based HT request classification to track the victim cache lines evicted by HT requests and measure its contribution to MT performance (good, bad and ugly) in FleximJ, similar to the mechanism proposed in [7];

2. Implemented the following five LLC replacement policies: (1) random, (2) LRU, (3) re-reference interval prediction based [5], (4) reuse distance prediction based and (5) reuse distance prediction with selective caching (or bypassing in essence) [6, 8], in FleximJ;

3. Obtained preliminary performance results of the above five LLC replacement policies for both the original and HT version of mst 1000 (via run-to-end detailed simulation), with their relative performance for the HT version as follows (left to right: lower performance to higher performance, the results for the original version will be given later):
random < LRU < reuse distance prediction < re-reference interval prediction < reuse distance prediction with selective caching;

4. Modified the simulated MESI coherence protocol to make possible L2 bypassing in a non-inclusive cache hierarchy by keeping shadow L1 tags in shared L2 (similar to the shadow tags mechanisms implemented in [9]). (the previously implemented CC protocol is for inclusive cache hierarchy, in which cache bypassing cannot be implemented);

5. Added basic implementation of MLP-aware hardware constructed p-thread chaining mechanism (optimization pending, MLP awareness pending) to help prefetch late HT requests, which is based on the idea proposed in [2].

2 Efficient Parallel Simulation of the Multicore Architecture on the Multicore Host[3]

1. Used actor based parallel programming model to implement the basic forms of both quantum based and slack based parallel simulation methodologies proposed in [1], with preliminary results indicating nontrivial speedups of simulation for multithreaded workloads on multiple

cores (optimization and speculative parallel simulation pending) . Its basic idea is to use one host thread (called core thread) for every simulated core (and its SMT threads and L1 I/D caches) and one host thread (called manager thread) for synchronizing the simulation of the core threads and the shared resources (such as the shared L2, on-chip network and DRAM controller);

2. Fixed many bugs in FleximJ to make it faster, more capable and more accurate (its current detailed simulation speed (without parallel simulation enabled) is about 100K instructions per second, which is comparable to the simulation speed of GEMS and is 3x faster than previously reported).

References

[1] Jianwei Chen, Lakshmi Kumar Dabbiru, Daniel Wong, Murali Annavaram, and Michel Dubois. Adaptive and speculative slack simulations of cmps on cmps. In *MICRO*, pages 523–534. IEEE, 2010.

[2] Jamison D. Collins, Dean M. Tullsen, Hong Wang, and John P. Shen. Dynamic speculative precomputation. In *Proceedings of the 34th Annual International Symposium on Microarchitecture*, pages 306–317, Austin, Texas, dec 1–5, 2001.

[3] Lieven Eeckhout. *Computer Architecture Performance Evaluation Methods*. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2010.

[4] Bruce L. Jacob, Spencer W. Ng, and David T. Wang. *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann, 2008.

[5] Aamer Jaleel, Kevin B. Theobald, Jr. Simon C. Steely, and Joel S. Emer. High performance cache replacement using re-reference interval prediction (rrip). In *Proc. 37th International Symposium on Computer Architecture (37th ISCA '2010)*, pages 60–71, Saint-Malo, France, jun 2010. ACM SIGARCH.

[6] Georgios Keramidas, Pavlos Petoumenos, and Stefanos Kaxiras. Cache replacement based on reuse-distance prediction. In *ICCD*, pages 245–250. IEEE, 2007.

[7] Bhavesh Mehta, Dana Vantrease, and Luke Yen. Cache showdown: The good, bad and ugly, 2004.

[8] Pavlos Petoumenos, Georgios Keramidas, and Stefanos Kaxiras. Instruction-based reuse distance prediction replacement policy, 2010.

[9] Hongzhou Zhao, Arrvindh Shriraman, and Sandhya Dwarkadas. Space: sharing pattern-based directory coherence for multicore scalability. In Valentina Salapura, Michael Gschwind, and Jens Knoop, editors, *PACT*, 19th International Conference on Parallel Architecture and Compilation Techniques (PACT 2010), Vienna, Austria, September 11-15, 2010, pages 135–146. ACM, 2010.