Improving Energy and Performance with Spintronics Caches in Multicore Systems

William Tuohy[1], Cong Ma[2], Pushkar Nandkar[2], Nishant Borse[2], and David J. Lilja[2]

[1] Department of Computer Science and Engineering, University of Minnesota - Twin Cities, Minneapolis MN 55455, USA
[2] Department of Electrical and Computer Engineering University of Minnesota - Twin Cities, Minneapolis MN 55455, USA

Abstract. Spintronic memory (STT-MRAM) is an attractive alternative technology to CMOS since it offers higher density and virtually no leakage current. Spintronic memory continues to require higher write energy, however, presenting a challenge to memory hierarchy design when energy consumption is a concern. Various techniques for reducing write energy have been studied in the past for a single processor, typically focusing on the last-level caches while keeping the first level caches in CMOS to avoid the write latency. In this work, use of STT-MRAM for the first level caches of a multicore processor is motivated by showing that the impact on throughput due to increased write latency is offset in many cases by increased cache size due to higher density. The Parsec benchmark suite is run on a modern multicore platform simulator, comparing performance and energy consumption of the spintronic cache system to a CMOS design. A small, fully-associative level-0 cache is then introduced (on the order of 8-64 cache lines), and shown to effectively hide the STT-MRAM write latency. Performance degradation due to write latency is restored or slightly improved, while cache energy consumption is reduced by 30-50% for 12 of the 13 benchmarks.

1 Introduction

As CMOS technology starts to face serious scaling and power consumption issues, the current SRAM designs become unable to meet the demand of big, fast and low power on-chip cache for multi-core implementations. A new technology, Spin-Transfer Torque-Magnetic RAM (STT-MRAM), one of the novel non-volatile memory family, has drawn substantial attention in recent years. STT-MRAM offers higher density than traditional SRAM cache, and its non-volatility facilitates low leakage power [13]. Also, STT-MRAM is one of few candidates that has almost the same read latency as current SRAM technology. With this higher cell density and low leakage power, STT-MRAM is generally considered as a viable potential alternative to SRAM in future on-chip caches.

STT-MRAM technology suffers from high dynamic energy consumption, however, due to high write current and longer write latency [16]. As others have shown, leakage power at the large last-level caches is the dominant energy consumer in CMOS cache hierarchies [1,9]. This work will consider a cache hierarchy

L. Lopes et al. (Eds.): Euro-Par 2014 Workshops, Part II, LNCS 8806, pp. 279–290, 2014.

with STT-MRAM last-level cache and CMOS first-level cache as the baseline configuration, and study the impact of converting the first-level cache to STT-MRAM as well. The impact on performance and energy over a range of write latencies is analyzed in detail. As would be expected, STT-MRAM reduces leakage power of the first-level cache, but increases energy consumed by processor writes significantly.

The write latency of STT-MRAM has the effect of reducing the available bandwidth into the cache, since this latency cannot be hidden with pipelining or other techniques. To address this fundamental limit, a small fully-associative level-0 (L0) cache was placed in front of the main L1 cache, similar to that proposed in [10]. This structure can be very small, yet have several benefits. By acting as a write-back cache, it absorbs processor writes at full bandwidth, and aggregates them into cache-line size writes to the STT L1 cache in the form of write backs. This technique improves bandwidth into the L1 cache, and can save energy if most of the processor writes can be absorbed there. Simulations show that performance lost due to high write latency can be recovered, while total cache energy consumption is reduced by 30% to 50% for 12 of the 13 benchmarks analyzed.

The contributions of this work include:

1. A detailed performance and energy-consumption study comparing a CMOS to a STT-MRAM first-level cache
2. An analysis of how well a small, fully-associative level-0 cache can overcome the performance degradation caused by long write latency of STT-MRAM
3. An energy comparison of dynamic energy consumed by the caches with and without the added level-0 cache

2 Experimental Methodology

Architectural simulations were performed with the gem5 simulator [4] running the Parsec benchmark suite [3]. A sampling technique similar to that described in SMARTS [22] was used to reduce simulation time while maintaining accuracy. All data is reported just for the parallel region-of-interest (ROI) using checkpoints compiled into the source by [5]. A four-processor system was simulated using a four-wide out-of-order execution model running at 2GHz. The first level cache is private to each CPU, while the last-level cache is shared. Cache coherence is enforced using the MESI protocol with inclusion. The sampled simulation data was verified by running complete simulations of the benchmark on selected configurations, and were found to be quite accurate. Table 1 lists the values of system parameters simulated for each benchmark. Every combination of these parameters was simulated. Cache read latency was assumed to be the same for CMOS and STT-MRAM technologies for the sizes utilized. The Parsec medium data set was used for full simulation runs due to runtime, so the sample runs use that data set as well.

To increase simulation throughput further, multiple simulations were run in parallel on non-overlapping regions of the program. Multiple checkpoints for

Table 1. Simulated Cache Configurations

Parameter	Values
L1 DCache Size	64K CMOS, 128K & 256K STT
STT L1 Write Latency	3ns, 5ns, 8ns
Cache Read Latency	L0 1 cycle, L1 3 cycles, L2 7 cycles (accessed sequentially)
L0 DCache Arrangement	baseline none; 512B, 1K, 4K fully-associative, private
L1 DCache Arrangement	2-way associative, private per CPU
L2 Cache Arrangement	4MB STT, 6ns write, 8-way associative, shared
Coherence Protocol	MESI with inclusion

each benchmark were created at 50 million cycle intervals using the simple atomic CPU model, starting at the ROI. The number of checkpoints created ranged from four for *canneal* to eighty-seven for *freqmine*. From each of these checkpoints, simulations were run in parallel, using GNU Parallel [19] to run thousands of small simulations in a relatively short period of time. Twenty-five samples were gathered from each checkpoint for every configuration simulated. The simulator was modified to allow switching between the simpler timing model CPU and the detailed out-of-order CPU model at different intervals. The simpler timing CPU was used to run the simulation forward for 900K cycles between periods of detailed simulation, keeping the caches and other dynamic structures active. The detailed out-of-order model was then switched in and run for 500K cycles, with the simulation statistics reset at each switchover. Performance impact was measured by comparing the instructions-per-cycle (IPC) of the benchmarks for the different configurations of cache size and write latency. The IPC of each benchmark was computed from the sampled set of IPCs of each interval simulated. Confidence intervals for 95% confidence were computed as well, using the techniques from [14]. In most cases the confidence intervals were very small so they do not change interpretation of the data.

Performance data and event counts relevant to dynamic cache energy consumption were gathered from the simulation statistics. Cacti [7] was modified to model the increased density of STT-MRAM devices, as well as the different leakage power and access transistor sizes required for different access times. A 32nm high-performance process was modeled by changing to bit-cell size from 146 F^2 to 40 F^2 with no leakage. Cacti is now able to model the fully associative caches used in this study. Parallel tag and data access was modeled for the L0 and L1 caches, while serial tag lookup was modeled for the L2. Conservative models were used for all values: L0 and L1 word write energy uses the line access value, since most of the energy is consumed in the peripheral circuitry rather than bit-cell access; array line loading still assumes a 6T cell array, in addition to the larger access transistor for STT. Access energy values include both tag and data array access, and leakage values include tag arrays and the different amounts of circuitry for the different cache sizes and arrangements. For STT-MRAM arrays, a per-bit energy of 300 fJ was added to the Cacti access energy for write operations, 64b for word writes and 512b for cache lines. Table 2 lists

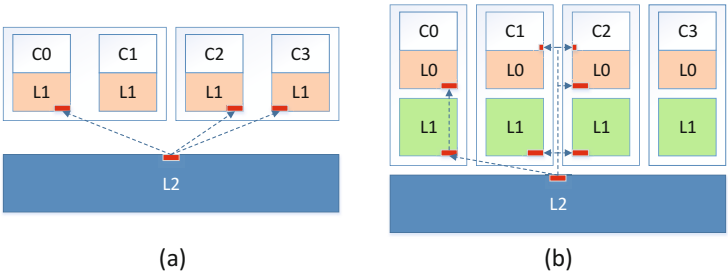

Fig. 1. The two-level hierarchy (a) evaluated in Sec. 3, with CMOS and STT-MRAM as the L1. Then a small fully-associative L0 is added (b) and evaluated in Sec. 4.

Table 2. Energy consumption parameters for the various cache structures

Structure	CMOS L1	STT L1		STT L2	CMOS Level-0		
size	64kB	128kB	256kB	4MB 8-way	512B	1kB	4kB
read (nJ)	0.032	0.033	0.062	0.385	0.0165	0.0165	0.0168
lineWrite(nJ)	0.055	0.220	0.230	0.290	0.0165	0.017	0.0203
wordWrite(nJ)	0.055	0.086	0.096	-	0.0165	0.017	0.0203
leakage(mW)	25	10.0	11.7	92	4.1	4.26	5.65

the power and energy parameters used to compute energy consumption from simulation activity.

3 Converting First Level from CMOS to STT-MRAM

Use of STT-MRAM at different levels of cache creates opposing performance effects. The increased write latency can degrade performance when this latency is exposed, while the larger caches enabled by higher density may increase performance for some programs. It has been seen that the increased latency at the L2 cache does not typically have much impact on performance due to a lower demand on the bandwidth [13]. The cache hierarchies evaluated in this paper are shown in Fig. 1. The configuration names and cache sizes listed in Table 3 are used in the figures and graphs that follow.

3.1 Performance Impact With L1 STT-MRAM

Typical sizes for L1 and L2 in CMOS are assumed to be 64kB for the L1 data cache, and 1MB for a shared L2 unified cache. For the STT-MRAM L2, a density increase of 4× is assumed, so a cache of 4MB would fit in roughly the same chip area and therefore have similar read latency, since read latency is a strong

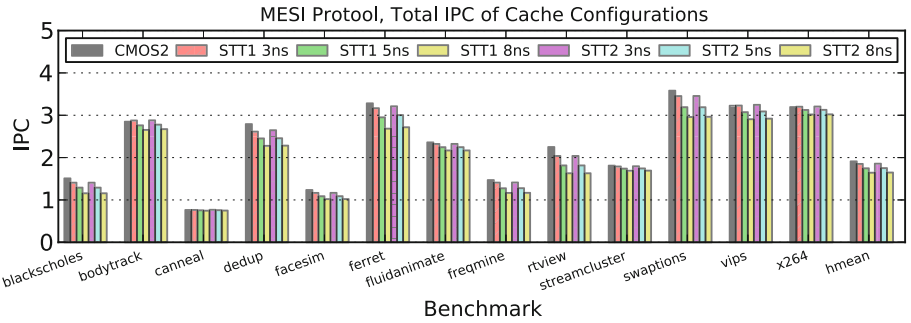

Fig. 2. Total IPC for Parsec benchmarks on a 4-CPU CMP, comparing CMOS2 (left-most bar) to STT1 and STT2 with varying write latency. Harmonic mean of IPC across all benchmarks is also shown, 1.9 for CMOS2. Mean IPC for STT1 is lower by 3% for 3ns write, 9% for 5ns, and 14% for 8ns writes. STT2 is virtually identical to STT1.

function of the length of wires and array size. For the L1 cache, while STT-MRAM density is still high, the actual cache design may be limited by the access transistor size rather than the magnetic tunnel junction device, so we model both 2× and 4× increase in L1 size. Since prior work [2] has shown little or no benefit from typical L3 cache sizes for the Parsec benchmarks, and because use of a third cache level would add more dynamic and static energy, we begin our analysis with a two-level cache hierarchy.

Figure 2 compares performance of CMOS2 to STT2 and STT1, with STT-MRAM write latencies ranging from 3ns to 8ns across all the Parsec benchmarks. The data is the sum of the individual core IPCs for a 4-CPU CMP, showing total system throughput. The best case was no slowdown, while the worst was about -27% for *rtview* at 8ns write latency. Eight of the thirteen benchmarks see less than about 5% slowdown at the 3ns STT-MRAM technology point. Recent work has shown experimental results achieving writes in the 1ns range [24,25], so it may not be overly optimistic to presume that STT-MRAM write latency in this range may become standard. At this design point, the worst slowdown among all benchmarks is less than 10%. There also appears to be little difference in performance between the two STT-MRAM L1 cache sizes considered, indicating that the larger L1 of STT2 does not provide significant improvement. Chip area may be better allocated to other features to improve throughput.

Table 3. Two-Level Hierarchy Configuration Names

Baseline Configuration Names	L1 Cache	L2 Cache
CMOS2	CMOS 64K	STT 4MB
STT1	STT 128K	STT 4MB
STT2	STT 256K	STT 4MB

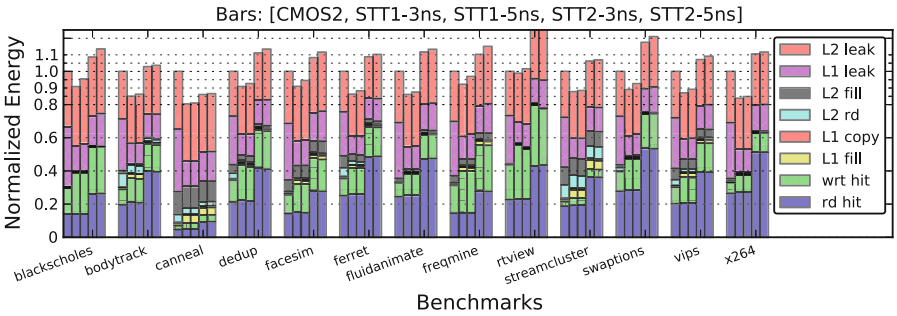

Fig. 3. Breakdown of energy consumption in the 2-level hierarchy, normalized to CMOS2

3.2 Energy Consumption With L1 STT-MRAM

Figure 3 shows the energy consumption of STT1 and STT2 cache configurations, normalized to CMOS2 (the leftmost bar of each benchmark). Though significantly reduced by implementing the L2 cache as STT-MRAM instead of CMOS, L2 leakage is still a large contributor to total cache energy consumption. L1 leakage (2nd segment from top) is also a large percentage of the total in the CMOS2 configuration. The STT1 configuration shows a drop in total energy compared to CMOS2, while the STT2 configuration causes an increase in total energy on most benchmarks due to higher dynamic read energy caused by the heavier loading of the internal array. Since performance was not improved with the larger L1 cache of STT2, L1 capacity does not appear to be the best use of chip area for these workloads. Dynamic energy of processor read and write hits to the L1 cache are the next significant cause of energy consumption in most benchmarks. The large amount of dynamic write hit energy indicates a significant amount of program data stores, which are likely the main cause of the performance drops seen in Figure 2. A technique to improve performance of writes to a high-latency structure, while reducing total energy consumption, would benefit the system in this case. One method to reduce both dynamic and static energy consumed by read and write operations is to use smaller structures, which also allow for faster access time. A fully-associative structure as small as 8 cache lines was added to the system to address these issues, with significant improvement observed.

4 Addition of Fully-Associative Level-0 Cache

The performance impact of STT-MRAM cache is due to reduced bandwidth into the cache stalling the processor. The portion of energy consumption due to processor writes that hit the L1 cache is a function of the number of writes in the program and the cache performance. To address both of these issues, the addition of a small fully-associative structure in front of the STT-MRAM cache

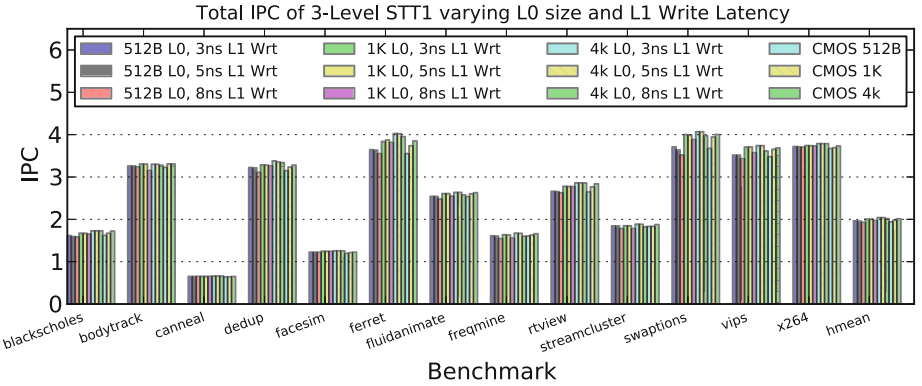

Fig. 4. Total IPC of benchmarks with L0 cache of various sizes, shown for the range of STT write latency from 3-8ns. The IPC drop seen in the two-level hierarchy with increasing write latency is not seen at the 3ns and 5ns configurations. The 8ns writes show a slight drop in some cases. The mean IPCs show no drop with added write latency.

is evaluated, similar to the scheme in [10]. This hierarchy is shown in Fig. 1b. A small structure can be fast enough to keep up with the processor when writes hit, and low enough energy to not offset the gains from implementing the larger L1 cache with STT-MRAM. By acting as a regular write-back cache, effective bandwidth to the L1 cache is increased by converting single-word writes into cacheline writes. By absorbing a high enough percentage of the processor writes, energy is reduced at the L1 cache since there are fewer write events. While at a much smaller scale, the goals are similar to the write aggregation schemes of mass storage systems [20,6].

Figure 4 shows the performance of the STT1 configuration with the added L0 cache of various sizes, ranging from 512B (8 lines) to 4KB (64 lines). As with the trends seen in Fig. 2, the difference in performance between STT1 and STT2 was negligible, so STT2 graphs are not shown. The drop in IPC with increased write latency is effectively eliminated in this scheme. In some benchmarks there is a small increase in performance with increased L0 size, on the order of 10% for *ferret* and *swaptions*, but most show little or no increase. Also shown, in the rightmost bars of each benchmark, is performance of a CMOS L1 of 64kB and the added L0 cache of the same three sizes to ensure a fair baseline. Performance was identical to or slightly below the STT1 performance for the same L0 size. This indicates that the L0 cache does hide the L1 write latency, while also allowing the larger L1 size to increase performance in some cases.

4.1 Energy Consumption With Level-0 Cache

Including the L0 cache in the hierarchy changes the number of events seen at the different cache levels. If enough high-energy events such as STT-MRAM

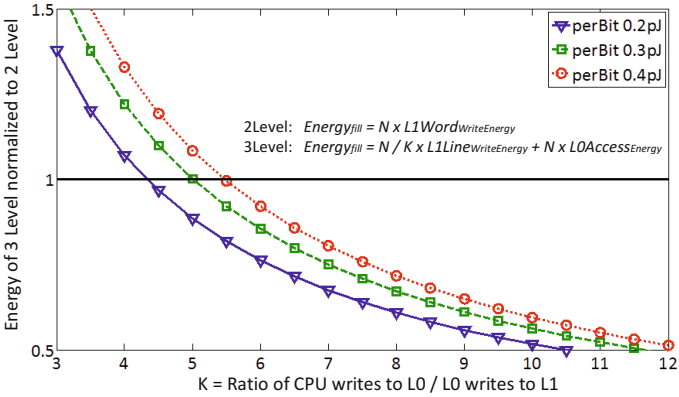

Fig. 5. Model of energy consumption in the L0 and L1 caches combined, as a function of the L1 STT-MRAM bit-cell write energy. Energy in the three-level (L0 + L1) is shown normalized to the two-level configuration (L1 only). The X-axis is the ratio of the number of CPU writes absorbed by the L0 to the number of lines written back from the L0 to the L1. As L1 bit-cell writes use less energy, relatively fewer writes need to be absorbed by the L0 to break even on energy.

writes are eliminated, the total energy consumed in the cache system is reduced. Figure 5 models the change in energy consumption as the effectiveness of the L0 in absorbing CPU writes changes. Portions of the energy equation that are not reduced, such as L0 write energy and leakage, create a lower limit and cause the slope to reduce farther to the right on the X-axis. The values in Table 2 use 0.3 pJ/bit for STT-MRAM write energy calculations.

Figure 6 shows the breakdown of energy consumption with the L0 cache in place. The two leftmost bars of each benchmark show the original CMOS and STT1 data of the two-level system, while the three rightmost bars show the different L0 cache sizes with a 128KB STT-MRAM L1 cache (STT1). While the motivation for use of the L0 cache was to reduce L1 dynamic write energy (the second segments from the bottom in each bar, respectively), L1 dynamic read energy also dropped significantly. The lowest two segments of the CMOS2 and STT1 bars are replaced by the low three segments of the L0 bars, which include L0 dynamic energy and L1 accesses. L0 leakage is also added, the third segment from the top. Examining the data for one benchmark, *blackscholes*, it can be seen that dynamic write energy (2nd segment from the bottom) grew significantly when going from CMOS2 to STT1. When the L0 was added, dynamic write energy was reduced to about the same magnitude as CMOS2 for the 512B case, and even further for the 1kB L0 case. The effectiveness in absorbing writes, K in Fig. 5, increased with the larger L0.

The size of the L0 does not change the total dynamic energy significantly for about half of the benchmarks. For others, such as *blackscholes, dedup*, and *freqmine*, a larger L0 further reduces L1 energy by reducing the L0 miss rate, as

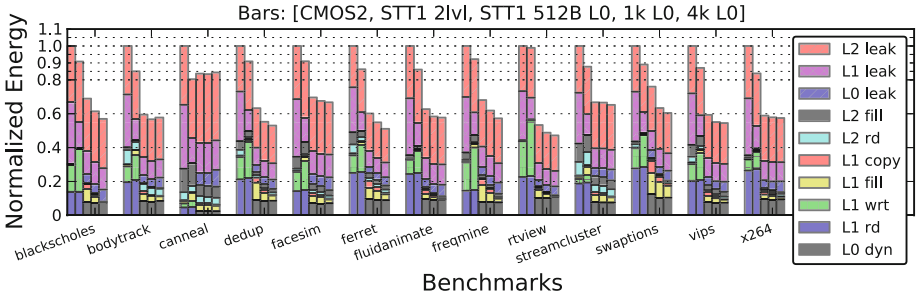

Fig. 6. Energy breakdown normalized to CMOS2 (two-level), showing CMOS2 and STT1 two-level, followed by STT1 with the three L0 cache sizes added. The lowest segments show L1 read for the two-level and L0 total dynamic energy for the 3-level; the 2nd segments show L1 write hits for the two-level and L1 line fills for the 3-level, one motivation for the L0 cache. Dynamic read energy is reduced significantly as well as write energy, indicating that the L0 is quite effective.

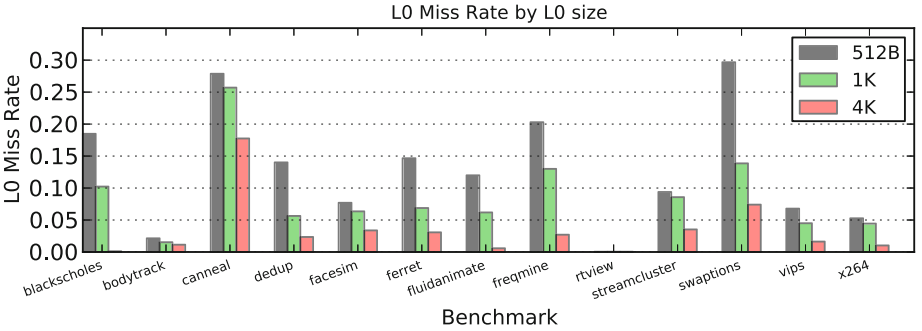

Fig. 7. Miss rate at the L0 cache for each benchmark as L0 size is increased. Bars for *blackscholes* at 4k and *rtview* are too small to show up at this scale.

shown in Fig. 7. Since the L0 modeled here is so small, it does not seem likely that the benchmark dataset size would make much difference in these results, but this should be verified with further simulation. Total energy consumption is reduced in the range of 30-50% from CMOS2 levels. For STT1 the gains are similar except for the *canneal* benchmark, which has poor locality and is more sensitive to memory latency than any other system parameter.

5 Related Work

To address the write power and latency problems, researchers have proposed several techniques: decreasing the retention time [16,17,9], modifying the cache hierarchy to use a mix of structures with different properties [13,23,11,17,21,8], implementing policies to limit write operations to high-power structures [15,26,12,18,1]. Decreasing the retention time trades reliability for

area and energy on a device level, requiring the use of timers and some form of data restoration or movement to reliable structures. Device or circuit-level techniques such as reduced retention time are orthogonal to our work; any technique that reduces write energy at the array or bit-cell level can work in conjunction with the cache hierarchies we have proposed. The various mixed structures are mostly focused on using STT-MRAM at the larger L2 and LLC structures. We are focused on getting maximum performance and energy efficiency when converting as much of the cache hierarchy as possible to STT-MRAM. The hybrid structures that utilize cache sets or ways with different properties require other hardware such as timers and predictors to decide where to allocate lines, and when to move lines from more volatile to less volatile areas. This extra data movement is overhead for energy consumption. Our scheme does not require any new hardware other than normal cache controllers and replacement logic.

6 Conclusions

The impact on performance and energy consumption of STT-MRAM use as first-level cache in a modern chip-multiprocessor has been evaluated, first with a standard two-level hierarchy and then with the addition of a small fully-associative structure. Performance with the STT-MRAM first-level cache was degraded in most benchmarks, and continued to degrade further as write latency was increased. Energy consumption of this configuration varied from a best-case of 20% reduction to worst-case of over 15% increase, since the various components such as leakage and write events changed significantly. Processor writes that hit the L1 cache became a high consumer of energy, more so than cache line fills and other events, in most benchmarks.

To address the performance penalty and potentially reduce energy consumption further, a small fully-associative structure was added to the system to act as a level-0 cache. When used as a standard write-back cache with no other additional hardware, a structure as small as eight cache lines was shown to be completely effective at eliminating the performance penalty, and in some cases enabled even higher performance to be realized by the larger STT-MRAM L1 cache. The total energy consumption of the cache hierarchy was reduced in the range of 30-50% for every benchmark except *canneal*, showing that STT-MRAM can be effectively used at the lower levels of the cache hierarchy when augmented with this small, fast structure to hide the write latency. The resulting system has no performance loss due to slow write operations, and in some cases speedups were observed, in addition to significant energy savings.

References

1. Ahn, J., Yoo, S., Choi, K.: Dasca: Dead write prediction assisted stt-ram cache architecture. In: 2014 IEEE 20th International Symposium on High Performance Computer Architecture, HPCA 2014 (February 2014)

2. Bhadauria, M., Weaver, V.M., McKee, S.A.: Understanding PARSEC performance on contemporary CMPs. In: IEEE International Symposium on Workload Characterization, IISWC 2009, pp. 98–107 (2009)
3. Bienia, C.: Benchmarking Modern Multiprocessors. Ph.D. thesis, Princeton University (January 2011)
4. Binkert, N., Beckmann, B., Black, G., Reinhardt, S.K., Saidi, A., Basu, A., Hestness, J., Hower, D.R., Krishna, T., Sardashti, S., Sen, R., Sewell, K., Shoaib, M., Vaish, N., Hill, M.D., Wood, D.A.: The gem5 simulator. SIGARCH Comput. Archit. News 39(2), 1–7 (2011), http://doi.acm.org/10.1145/2024716.2024718
5. Gebhart, M., Hestness, J., Fatehi, E., Gratz, P., Keckler, S.W.: Running parsec 2.1 on m5. Tech. rep., The University of Texas at Austin, Department of Computer Science (October 2009)
6. Gill, B.S., Modha, D.S.: Wow: Wise ordering for writes - combining spatial and temporal locality in non-volatile caches. In: Proceedings of the 4th Conference on USENIX Conference on File and Storage Technologies, FAST 2005, vol. 4, p. 10. USENIX Association, Berkeley (2005)
7. Hewlett-Packard Development Company, L.: Cacti 6.5 (2009), http://www.hpl.hp.com/research/cacti/
8. Jadidi, A., Arjomand, M., Sarbazi-Azad, H.: High-endurance and performance-efficient design of hybrid cache architectures through adaptive line replacement. In: ISLPED 2011: Proceedings of the 17th IEEE/ACM International Symposium on Low-Power Electronics and Design. IEEE Press (August 2011)
9. Jog, A., Mishra, A.K., Xu, C., Xie, Y., Narayanan, V., Iyer, R.K., Das, C.R.: Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs. In: DAC 2012: Proceedings of the 49th Annual Design Automation Conference, pp. 243–252 (2012)
10. Jouppi, N.P.: Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers. ACM SIGARCH Computer Architecture News 18, 364–373 (1990)
11. Kim, Y., Gupta, S.K., Park, S.P., Panagopoulos, G., Roy, K.: Write-optimized reliable design of STT MRAM. In: ISLPED 2012: Proceedings of the 2012 ACM/IEEE international symposium on Low Power Electronics and Design. ACM Request Permissions (July 2012)
12. Kwon, K.W., Choday, S.H., Kim, Y., Roy, K.: AWARE (Asymmetric Write Architecture With REdundant Blocks): A High Write Speed STT-MRAM Cache Architecture. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 22(4), 712–720
13. Park, S.P., Gupta, S., Mojumder, N., Raghunathan, A., Roy, K.: Future cache design using STT MRAMs for improved energy efficiency: devices, circuits and architecture. In: DAC 2012: Proceedings of the 49th Annual Design Automation Conference. ACM Request Permissions (June 2012)
14. Patil, S., Lilja, D.J.: Using resampling techniques to compute confidence intervals for the harmonic mean of rate-based performance metrics. Computer Architecture Letters 9(1), 1–4 (2010)
15. Rasquinha, M., Choudhary, D., Chatterjee, S., Mukhopadhyay, S., Yalamanchili, S.: An energy efficient cache design using spin torque transfer (STT) RAM. In: ISLPED 2010: Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design. ACM Request Permissions (August 2010)

16. Smullen, C.W.I., Mohan, V., Nigam, A., Gurumurthi, S., Stan, M.R.J.: Relaxing Non-Volatility for Fast and Energy-Efficient STT-RAM Caches. In: 2011 IEEE 17th International Symposium on High Performance Computer Architecture (HPCA), pp. 50–61 (2011)
17. Sun, Z., Bi, X., Li, H.H., Wong, W.F., Ong, Z.L., Zhu, X., Wu, W.: Multi retention level STT-RAM cache designs with a dynamic refresh scheme. In: MICRO-44 2011: Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture. ACM Request Permissions (December 2011)
18. Sun, Z., Li, H., Wu, W.: A dual-mode architecture for fast-switching STT-RAM. In: ISLPED 2012: Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design. ACM Request Permissions (July 2012)
19. Tange, O.: Gnu parallel - the command-line power tool. ;Login: The USENIX Magazine 36(1), 42–47 (2011), `http://www.gnu.org/s/parallel`
20. Varma, A., Jacobson, Q.: Destage algorithms for disk arrays with non-volatile caches. In: Proceedings of the 22nd Annual International Symposium on Computer Architecture, pp. 83–95 (June 1995)
21. Wu, X., Li, J., Zhang, L., Speight, E., Xie, Y.: Power and performance of read-write aware hybrid caches with non-volatile memories. In: Design, Automation Test in Europe Conference Exhibition, DATE 2009, pp. 737–742 (April 2009)
22. Wunderlich, R.E., Wenisch, T.F., Falsafi, B., Hoe, J.C.: SMARTS: accelerating microarchitecture simulation via rigorous statistical sampling. In: ISCA 2003: Proceedings of the 30th Annual International Symposium on Computer Architecture. ACM (June 2003)
23. Xu, W., Sun, H., Wang, X., Chen, Y., Zhang, T.: Design of last-level on-chip cache using spin-torque transfer ram (stt ram). IEEE Transactions on Very Large Scale Integration (VLSI) Systems 19(3), 483–493 (2011)
24. Yoda, H., Fujita, S., Shimomura, N., Kitagawa, E., Abe, K., Nomura, K., Noguchi, H., Ito, J.: Progress of STT-MRAM technology and the effect on normally-off computing systems. In: 2012 IEEE International Electron Devices Meeting (IEDM), pp. 11.3.1–11.3.4 (2012)
25. Zhao, H., Glass, B., Amiri, P.K., Lyle, A., Zhang, Y., Chen, Y.J., Rowlands, G., Upadhyaya, P., Zeng, Z., Katine, J.A., Langer, J., Galatsis, K., Jiang, H., Wang, K.L., Krivorotov, I.N., Wang, J.P.: Sub-200 ps spin transfer torque switching in in-plane magnetic tunnel junctions with interface perpendicular anisotropy. Journal of Physics D: Applied Physics 45(2), 025001 (2011)
26. Zhou, P., Zhao, B., Yang, J., Zhang, Y.: Energy reduction for STT-RAM using early write termination. In: ICCAD 2009: Proceedings of the 2009 International Conference on Computer-Aided Design. ACM Request Permissions (November 2009)