DESIGNING LARGE HYBRID CACHE FOR FUTURE HPC SYSTEMS

Jiacong He
Department of Electrical Engineering
The University of Texas at Dallas
800 W Campbell Rd, Richardson, TX, USA
Email: jiacong.he@utdallas.edu

Joseph Callenes-Sloan
Department of Electrical Engineering
The University of Texas at Dallas
800 W Campbell Rd, Richardson, TX, USA
Email: jcallenes.sloan@utdallas.edu

ABSTRACT

DRAM cache is a large cache stacked on the processors die using 3D-stacking technology, which may be used in the future High-Performance Computing (HPC) systems to reduce latency and increase bandwidth. However, the energy becomes an inevitable challenge with the increasing cache capacity. In this paper, we first propose a large hybrid cache for future HPC systems, which can effectively reduce the static energy compared with the DRAM cache. Further, we apply volatile STT-RAM as part of the hybrid cache to reduce both the static and dynamic energy of the DRAM cache. Finally, we propose to maintain the cache tag array in the region of the hybrid cache with less read latency to improve performance. Experimental results show our hybrid cache reduces energy by 31.6% and improves performance by 18.8% on average.

Keywords: DRAM cache, STT-RAM, performance, energy, HPC.

1 INTRODUCTION

Future high-performance computing demands large cache capacity and high memory bandwidth. However, the existing SRAM cache with low density hinders the increment of cache capacity, and the limited pin count leads to the memory bandwidth wall. Recently, the 2.5D/3D die-stacking technologies are widely used in the major processor vendors (e.g., Intel Xeon Phi Processor includes up to 16GB 2.5D-stacking DRAM memory). Also, DRAM has already been used as a large cache in commercial supercomputer (e.g., IBM POWER8 uses up to 128MB eDRAM as L4 cache per socket). Thus, many researchers (Loh and Hill 2011, Huang and Nagarajan 2014, Zhao et al. 2007) proposed to use 3D die-stacking DRAM as a last-level cache to increase the cache capacity and off-chip memory bandwidth. The DRAM cache consists of multiple layers of DRAM stacked on the processor die using Through-Silicon via (TSV). It is potential to meet the workloads demand of future HPC systems by increasing the on-chip cache capacity up to gigabytes of storage and providing orders of magnitude higher bandwidth.

However, die-stacking DRAM cache with large capacity suffers from high leakage power and becomes increasingly susceptible to error due to the process scaling. Also, the 3D design has more challenge in power and thermal management because multiple stacking layers result in higher power densities. Recently, Spin-Transfer Torque RAM (STT-RAM), as an emerging non-volatile memory technology, is potential to be used as a large cache due to its near-zero leakage and high density. However, STT-RAM has the disadvantage of high write energy and high write latency, so it can not directly substitute for DRAM cache without any optimizations. Thus, STT-RAM is commonly used in a hybrid cache to utilize the advantage of different memory technologies. For example, recent works (Li et al. 2011, Wu et al. 2009) leveraged non-volatile

SpringSim-HPC 2017, April 23-26, Virginia Beach, VA, USA

STT-RAM to build a hybrid cache with SRAM. However, there are fabrication challenges of the hybrid cache in the conventional 2D design. We observe the emerging 3D stacking technology provides an excellent opportunity to build a large hybrid cache by integrating different wafers with different memory techniques.

While today's servers need tens to hundreds of gigabytes of DRAM each, the corresponding demand for die-stacked cache capacity varies between hundreds of megabytes to several gigabytes (Jevdjic et al. 2013). Thus, our proposed large hybrid cache also requires large tag storage (64MB tag array for 1GB cache) considering to use conventional 64B cache block. Ideally, the tag array should be stored in the SRAM cache to make the tag access latency as small as possible, while it is impractical due to the precious SRAM cache capacity. Some researchers proposed to store the tag array within the large on-die cache, which needs to optimize the latency of tag access by adding extra design complexity. We notice that the tag management policy in the large hybrid cache is also important, but previous works (Cong et al. 2011, Li et al. 2011, Wu et al. 2009) rarely considered this issue.

In this paper, we propose a large last-level hybrid cache for HPC systems, which consists of DRAM and STT-RAM regions. Each region can be composed of several layers stacked upon each other. The DRAM layers are used as the main component of the hybrid cache due to their high endurance. The STT-RAM layers with small leakage are used to reduce the static energy consumption of the hybrid cache. Also, we notice the STT-RAM can be relaxed (Smullen et al. 2011) to reduce its high write energy and latency by sacrificing its non-volatility. Thus, the non-volatile STT-RAM in the hybrid cache is replaced by volatile STT-RAM to further reduce both static and dynamic energy of the hybrid cache. Finally, we observe that there are two different tag array in the DRAM region and STT-RAM region of the hybrid cache respectively. And the read latency in these two regions is unbalanced due to the disparate memory technologies, thus we propose to move all tag array to the cache region (DRAM or STT-RAM) with lower read latency to improve the performance.

Overall, our contributions are as follows.

- We propose a large hybrid cache for future HPC systems to reduce static energy.
- We use volatile STT-RAM as part of the hybrid cache to reduce both static and dynamic energy.
- We optimize the tag management of the proposed hybrid caches to improve performance.

2 MOTIVATION

With the increasing frequency of CPU, more and more programs will be limited in the performance by the systems' memory bandwidth, rather than by the computational performance of the CPU. Also, the high-end computer spends over 90% of their time idle waiting for cache misses and fetching data from off-chip memory. Thus, the conventional DRAM memory with low bandwidth and high latency leads to the memory wall problem in the current shared-memory HPC systems. To handle the memory wall problem in HPC systems, die-stacking technologies have recently drawn much attention from the research community as a viable solution. Through die-stacking technology, DRAM can be stacked on top of the processor die (3D) or on a separate die connected through-silicon interposer (2.5D), providing an additional cache capacity to the traditional SRAM cache, much higher bandwidth and lower interconnect latency compared to off-chip DRAM memory.

Further, a typical supercomputer consumes prohibitively large amounts of electrical power for computing, while the demand for computing is increasing exponentially as a consequence of data explosion in the scientific computing. It is observed that Last Level Cache (LLC) has become a significant source of both static and dynamic energy consumption in modern processors, consuming up to 17% of total core energy. Thus, although DRAM cache as LLC can be used to alleviate bandwidth and latency pressure in the HPC systems, the large on-die cache exacerbates the energy challenge. To handle the power wall challenge in the

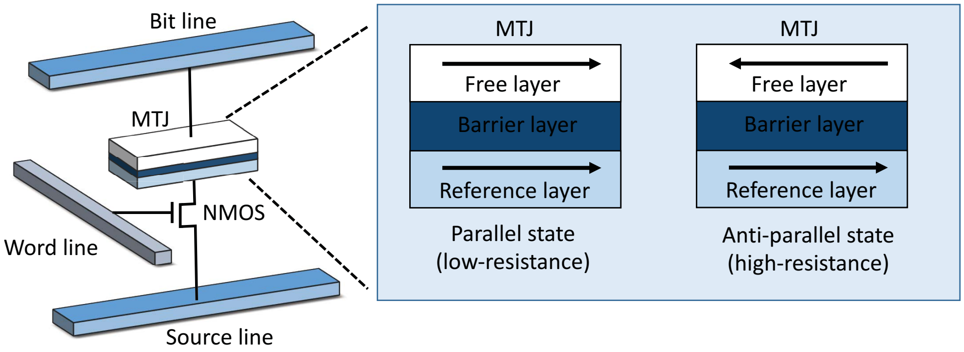


Figure 1: The structure of STT-RAM cell with 1MTJ 1T.

conventional cache, emerging non-volatile STT-RAM is being explored as potential alternatives of SRAM and eDRAM cache. A typical cell size of STT-RAM is 40F^2 compared with 146F^2 SRAM cell size, so STT-RAM is attractive to be used as a large LLC cache for area savings. Also, The read latency and energy of STT-RAM are comparable to SRAM and DRAM, and STT-RAM has near-zero leakage power and zero refresh energy. However, it is also noticed the STT-RAM is 2x worse in write latency and 10x worse in write energy compared with DRAM. Thus, STT-RAM can not be directly used to replace DRAM as a large cache based on the write-intensive nature of many scientific workloads.

To fully utilize the benefit of different memory techniques, we observe the large hybrid cache consisting of STT-RAM and DRAM can effectively reduce overall energy while maintaining performance at an efficient level. However, one key challenge in designing a large on-die cache is the cache tag management. We observe that the read latency is unbalanced in the hybrid cache, and the tag array access is actually a read operation, which can be utilized to optimize the hybrid cache performance by moving tag array to the cache region (DRAM or STT-RAM) with lower read latency.

3 BACKGROUND

STT-RAM. As shown in the Figure 1, the STT-RAM cell has an access transistor that connects the storage device and the bitline. It also has a Magnetic Tunnel Junction (MTJ) to store binary data, and the MTJ consists of two ferromagnetic layers and one tunnel barrier layer. The resistance of the MTJ is used to represent the binary data stored in the cell, which is determined by the relative magnetization direction of these two layers (Kultursay et al. 2013). And the low and high resistance are used to represent logical 0 and 1 respectively. Further, the data retention time of STT-RAM could be relaxed to reduce its high write energy by shrinking the planar area of the MTJ or decreasing the thickness of the free layer.

DRAM Cache. Previous researchers divide the DRAM cache into two categories. One is the block-based DRAM cache, which is architected as a large, software-transparent last-level cache (Loh and Hill 2011). It uses the 64B block size of conventional SRAM cache to optimize temporal locality. Thus, it requires a large amount of space for tag storage (16MB tag storage for 256MB DRAM cache). Another is the page-based DRAM cache design using a much larger cache block size of 2KB to 4KB (Jevdjic et al. 2014, Jevdjic et al. 2013). The tag overhead in block-based design is reduced to a few megabytes. However, a large cache line may fetch many unused data on a DRAM cache miss from the low-bandwidth off-chip memory.

Hybrid Cache. Different memory technologies have different characteristics of power, performance, and density. Hybrid cache integrates different memory technologies and achieves the overall optimal design. There are two types of hybrid cache architectures (Wu et al. 2009), one is the inter-cache design (every cache level of the cache hierarchy has disparate memory technologies) and the other is intra-cache design (single

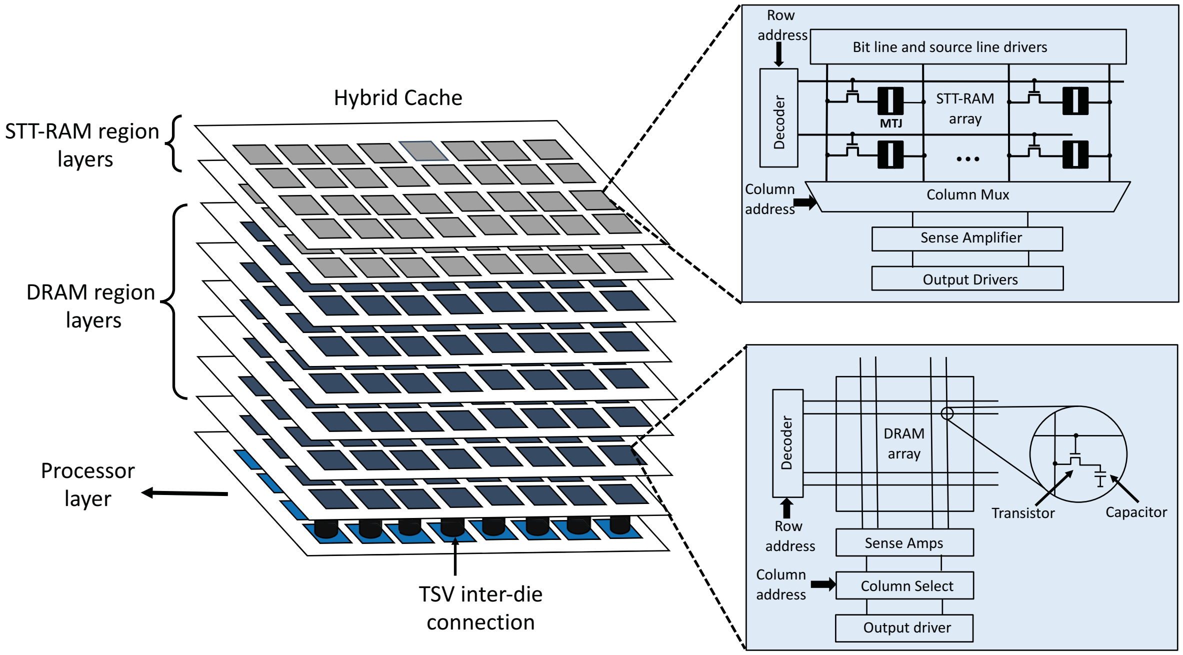


Figure 2: Proposed 3D-stacking hybrid cache architecture overview, with DRAM and STT-RAM cell array.

cache level consists of different memory technologies, and this is what our paper focuses on). Previously, many researchers proposed hybrid SRAM/STT-RAM cache and hybrid SRAM/DRAM cache (Wang et al. 2014, Cong et al. 2011), but little work discussed hybrid DRAM/STT-RAM cache. Different from prior work (He and Callenes-Sloan 2016), we focus on the evaluation of hybrid cache for the scientific and large-scale applications, especially based on the HPC systems.

4 HYBRID CACHE FOR FUTURE HPC SYSTEMS

4.1 Architectural Design Overview

We propose a large die-stacking hybrid cache consisting of DRAM region and STT-RAM region. The total hybrid cache capacity is 1GB, where DRAM region capacity is 768MB and STT-RAM region capacity is 256MB. The DRAM region is divided into 6 layers with 128MB per layer, and the STT-RAM region is divided into 2 layers with 128MB per layer. Every layer is stacked upon each other as illustrated in the Figure 2. The lowest die is the processors die containing 32 cores, each core has private L1 cache (64KB per core), L2 cache (4MB per core) cache and shared L3 cache (8MB). The DRAM region with 6 layers is stacked on the processors die, providing cache access performance due to shorter vertical wire connection. The STT-RAM region with 2 layers is stacked on the DRAM region providing static energy reduction. Once there is a data request from the CPU, the request is transmitted from the processors' layer to the last layer of the hybrid cache using low-latency TSV inter-die connection. The DRAM region is first accessed to buffer a large amount write request due to its large capacity, which can help to alleviate the high write energy pressure in the STT-RAM region.

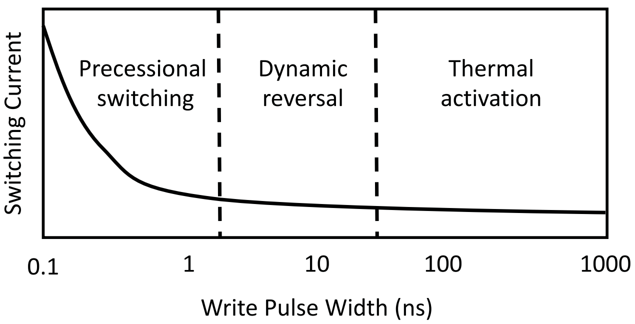


Figure 3: Three different working regions of MTJ.

Table 1: Relation between retention time and thermal factor at 278K.

Retention time	10 years	1 year	1 month	1 week	1 day	1 hour	1 min	1 s	10 ms
Δ	40.12	37.56	35.34	33.96	32.02	38.65	24.47	20.32	15.78

4.2 STT-RAM Optimization

Although emerging STT-RAM can efficiently reduce the static energy compared with conventional DRAM, its high write energy and latency still pose a large challenge in using STT-RAM in the large cache. In this paper, we propose to use disparate memory technologies (DRAM and STT-RAM) to build a large hybrid cache based on the 3D-stacking technique, which can effectively reduce both the static and dynamic energy in the large cache, and keep the original performance benefit. However, the HPC workloads and scientific applications usually are write-intensive with large datasets, so the STT-RAM used in the hybrid cache still need to be optimized to accommodate data-intensive workloads. It is noticed that the non-volatility of STT-RAM can be sacrificed to reduce its high write energy and latency. Thus, we optimize our hybrid cache by replacing non-volatile STT-RAM with volatile STT-RAM to achieve better energy and performance efficiency. In the following parts, we first analyze the non-volatility characteristics of STT-RAM, and then we show how to optimize hybrid cache using volatile STT-RAM in terms of energy and performance.

4.2.1 STT-RAM Non-volatility

Figure 1 depicts the structure of the STT-RAM cell array, where the STT-RAM cell is connected to word line (WL), bit line (BL) and source line (SL). The WL is used to select the specific row, and the voltage difference between SL and BL is used to complete write and read operation. When executing a read operation, a negative voltage is applied between SL and BL, and the current flowing through the free layer of the MTJ is sensed by the sense amplifier. To write data to an MTJ, a large current must be pushed through the MTJ to change the magnetic orientation of the free layer. Depending on the direction of the current, the free layer becomes parallel or anti-parallel to the fixed layer. The amount of current required for writing into an MTJ should be larger than a critical current.

MTJ has three regions, including the thermal activation region, dynamic reverse region and processional switching region (Diao et al. 2007). Their distribution is shown in the Figure 3, and the required switching current in each working region can be calculated by:

$$J_C^{THM}(T_{sw}) = J_{C0}(1 - \frac{1}{\Delta} ln(\frac{T_{sw}}{\tau_0})) \qquad (T_{sw} > 10ns) \qquad (1)$$

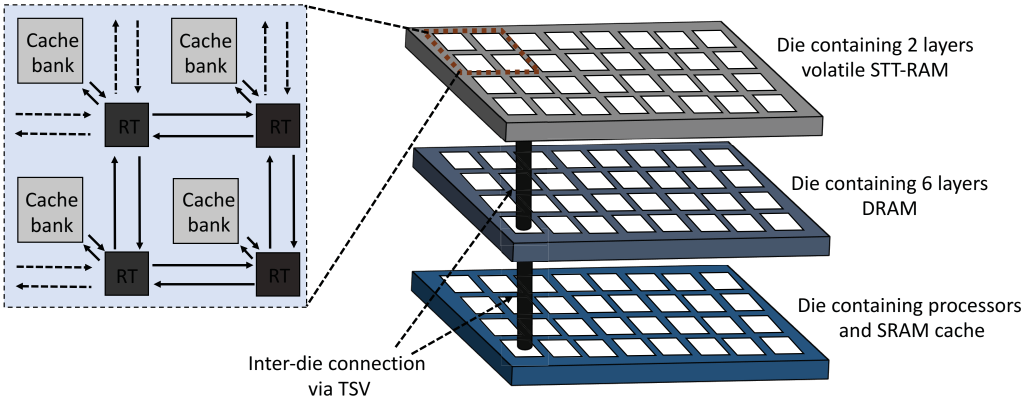

Cache bank
Cache bank
RT
RT
Cache bank
Cache bank
RT
RT
Die containing 2 layers volatile STT-RAM
Die containing 6 layers DRAM
Die containing processors and SRAM cache
Inter-die connection via TSV

Figure 4: 3D Hybrid cache including 1 volatile STT-RAM die, 1 DRAM die and 1 processors die. There are 32 processing cores, 32 cache banks per cache layer. Cache banks are connected through NoC routers.

$$J_C^{DYN}(T_{sw}) = \frac{J_C^{THM}(T_{sw}) + J_C^{PREC}(T_{sw})e^{-A(T_w - T_{PIV})}}{1 + e^{-A(T_w - T_{PIV})}} \qquad (10ns \geq T_{sw} > 3ns) \tag{2}$$

$$J_C^{PREC}(T_{sw}) = J_{C0} + \frac{ln(\frac{\pi}{2\theta})}{T_{sw}} \qquad (T_{sw} \leq 3ns) \tag{3}$$

where $J_C(T_w)$ is the required switching current density, J_{CO} is the threshold of the switching current density, T_{sw} is the switching pulse width, τ_0 is the relaxation time, Δ is the thermal stability of MTJ. The thermal stability of MTJ determines the retention time T_{ret} of STT-RAM (Diao et al. 2007), which can be modeled as: $T_{ret} = \frac{1}{f_0}e^{\Delta}$. Based on the analysis above we estimate the average time for MTJ bits flip. In the Table 1, we show the retention time changing with thermal factors at temperature 278K. Reducing the size of MTJ leads to shorter retention time, which provides larger storage density and less write energy of STT-RAM.

4.2.2 Hybrid Cache with Volatile STT-RAM

To design a hybrid cache with volatile STT-RAM, we need to first determine a suitable data retention time for the STT-RAM. As shown in Table 1, the retention time varies from year to millisecond, and lower retention time needs extra refresh operation like DRAM to keep data valid. On the one hand, if the retention time is selected too long, the high write energy and latency of STT-RAM can not be effectively reduced. On the other hand, if the retention time is selected too short, the refresh operation of volatile STT-RAM will lead to high refresh energy consumption. In order to choose an appropriate retention time that can balance the high write energy and extra refresh energy of the STT-RAM, we observe that the data refresh period of on-die DRAM can be used as a good reference for determining the retention time of STT-RAM.

It is known the refresh period of commodity DRAM is 64ms, which means the DRAM restores the degraded voltage stored in the DRAM cell capacitors for every 64ms due to DRAM volatile nature. For a detailed description of DRAM refresh, we refer the reader to (Liu et al. 2012). However, the refresh period of on-die DRAM cache is smaller than 64ms due to the usage of fast logic transistors, which have higher leakage than the DRAM memory. To reduce the design complexity of refresh circuit, the retention time of volatile STT-RAM should be close to the refresh rate of DRAM cache. Therefore, we conduct an application-driven study to analyze the refresh times of the DRAM cache blocks to determine a suitable data retention time. An extensive analysis of emerging workloads indicates that the average retention times for the DRAM cache

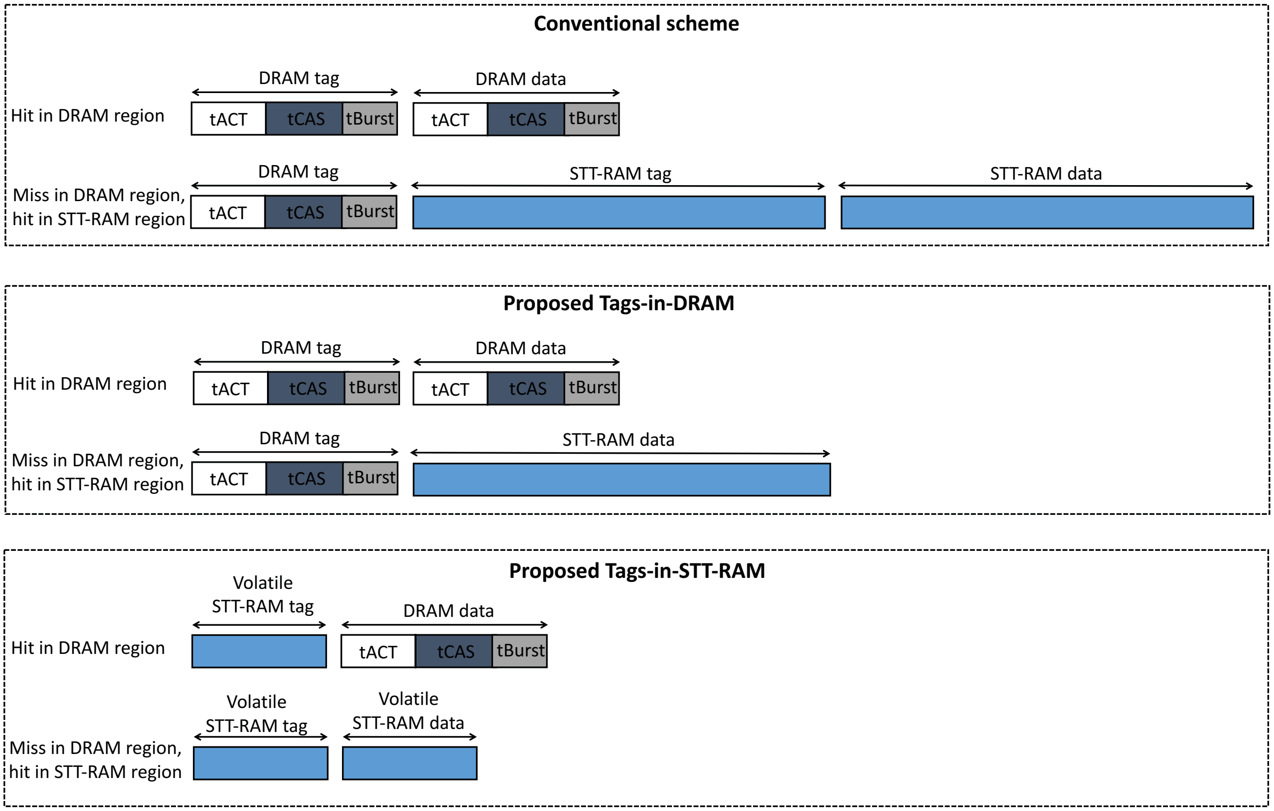


Figure 5: Access latency of different types of hybrid cache architecture.

blocks is close to 100μs. Although this aggressive refresh rate used in the STT-RAM will lead to over 95% write energy reduction, while the refresh energy is increased over 10x compared with 64ms retention time. To balance the write energy reduction and the overhead of refresh energy increment, we advocate the retention time of STT-RAM to be 0.5ms, which has about 90% write energy reduction and 5x refresh energy increment.

As shown in the Figure 4, our hybrid cache uses TSV as the vertical interconnection between hybrid cache and processor cores. In each core, there is a hybrid cache controller connected to the hybrid cache, from which data and request are moved through layers between processing cores and caches. Each core within the 2D processor layer is communicated through Network-on-Chip (NoC) routers. Also, the latency for traversing each layer is negligible compared to that between two NoC routers. Each layer of the hybrid cache is divided into 32 banks, and several cache banks in each layer which are connected with NoC routers.

4.3 Tag Management for the Large Hybrid Cache

Base on the hybrid cache architecture, we observe that the conventional tag array of the large hybrid cache actually consists of two parts, one is the tag array in the DRAM region and the other is the tag array in the STT-RAM region. Due to the unbalanced read latency of the disparate memory technologies, the latency to access tag array is also unbalanced. Different from conventional design based on NUCA (Das et al. 2015) and NUMA (Li et al. 2013), we consider the unbalanced read latency of the two proposed hybrid cache architectures, which are the hybrid cache with non-volatile STT-RAM and with volatile STT-RAM respectively.

Table 2: System Configurations.

CPU core	32-core OoO, 3.2GHz, 16-core/socket
SRAM Cache	L1: 64KB, 8-way, 4-cycle load-to-use, 64B linesize L2: 4MB, 16-way, 15-cycle hit latency, sequential tag/data access L3: 8MB, 32-way, 64B linesize, LRU, write-back, write-allocate policy
Hybrid LLC	(H1) STT-RAM: 256MB, 29-way, 1.13/21.35 pJ/bit for R/W energy (H2) Volatile STT-RAM: 256MB, 29-way, 1.06/1.18 pJ/bit for R/W energy DRAM: 768MB, 29-way, 1.25/1.31 pJ/bit for R/W energy
Off-Chip DRAM	2 channels, 2 ranks per channel, 8 banks per rank, DDR3-1600 (12.8GB/s)
Network Parameter	9 layers, 32 TSVs, 2-cycle router latency

For the hybrid cache with non-volatile STT-RAM, it is noticed the read latency of STT-RAM is 2x higher than the DRAM. Thus, we propose to move the tag array of STT-RAM region to the DRAM region (Tags-in-DRAM). Similarly, for the hybrid cache with volatile STT-RAM, the volatile STT-RAM is optimized to have lower read latency than the DRAM (over 20%) by shrinking the thickness of MTJ. Thus, we propose to move the tag array of DRAM region to the STT-RAM region (Tags-in-STT-RAM). The reason behind these designs is that the tag access actually is a read operation (to determine cache hit or miss), and we always move the tag array to the region of the hybrid cache with lower read latency to improve overall performance. In the conventional tag management shown in the Figure 5, high latency is wasted in the tag access in the separate hybrid cache regions. Compared to the conventional scheme, Tags-in-DRAM and Tags-in-STT-RAM design can reduce the tag access latency by obviating the need to access the high-latency tag array.

5 EXPERIMENTAL METHODOLOGY

We extend gem5 simulator (Binkert et al. 2011) to simulate a 32-core system with 3-level SRAM cache and a last-level hybrid cache. The STT-RAM region and DRAM region are modeled similar to (Kultursay et al. 2013) and (Loh and Hill 2011) respectively. The major system parameters are listed in the Table 2. All the experimental parameters of 3D volatile/non-volatile STT-RAM and DRAM cache are obtained from the modified version of DESTINY (Poremba et al. 2015). McPAT (Li et al. 2009) is used to get the power values of the hybrid cache. The state-of-the-art DRAM cache (Loh and Hill 2011) is used as our baseline. The 3D TSV model parameter is based on (Sun et al. 2009). The simulations were done for 2 configurations: **(H1)** The hybrid cache with non-volatile STT-RAM; **(H2)** The hybrid cache with volatile STT-RAM.

We analyze a set of large-scale HPC applications from the PARSEC (Bienia et al. 2008) and scientific applications from the SPEC CPU2006 (Henning 2006) to evaluate the energy and performance of the hybrid cache. For each of the workloads, we warmed up the simulation for one billion cycles and collected results for one billion cycles. For the evaluation metrics, we use energy savings and IPC speedups to show how much energy and performance efficiency can be achieved.

PARSEC benchmark focuses on emerging workloads and was designed to be representative of next-generation shared-memory programs for chip-multiprocessors. It consists of computationally intensive and parallel programs that are very common in the domain of HPC.

SPEC CPU2006 benchmark is comprised of various scientific and real-life applications, which are used to measure the computer performance stressing on the system's processor and memory subsystem. It evaluates performance by measuring how fast the computer completes a single task.

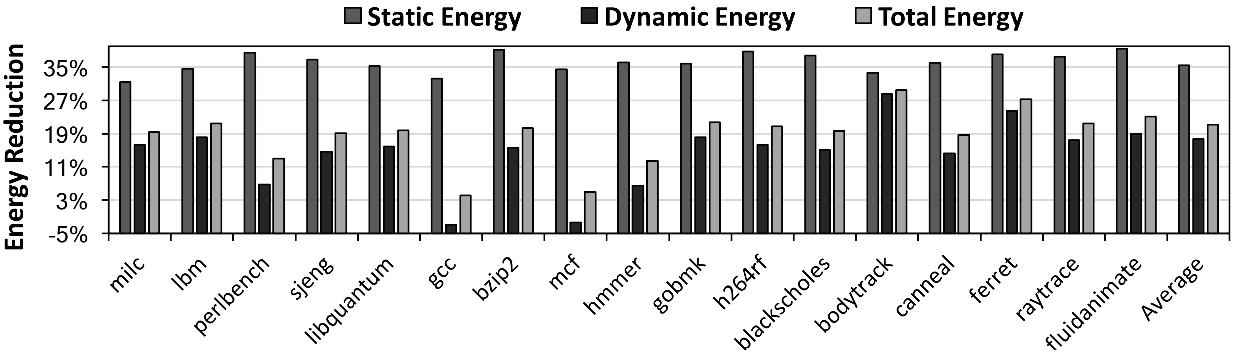


Figure 6: Energy reduction of hybrid cache with non-volatile STT-RAM (H1).

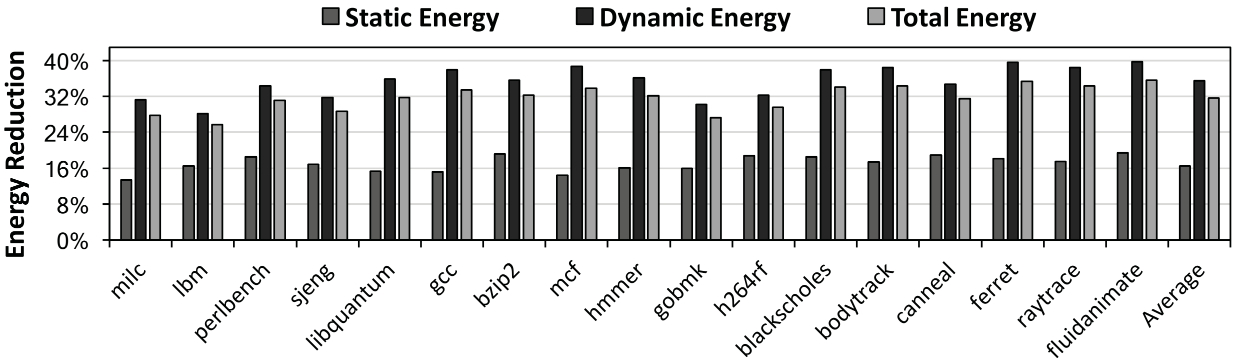


Figure 7: Energy reduction of hybrid cache with volatile STT-RAM (H2).

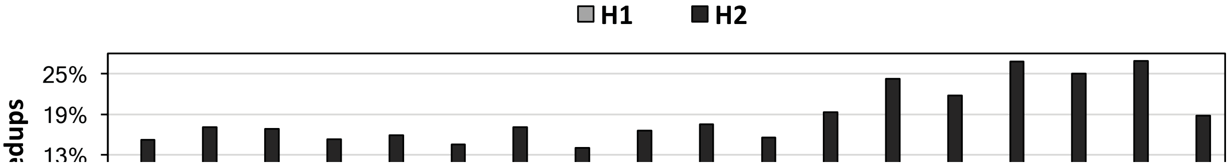
6 RESULTS AND ANALYSIS

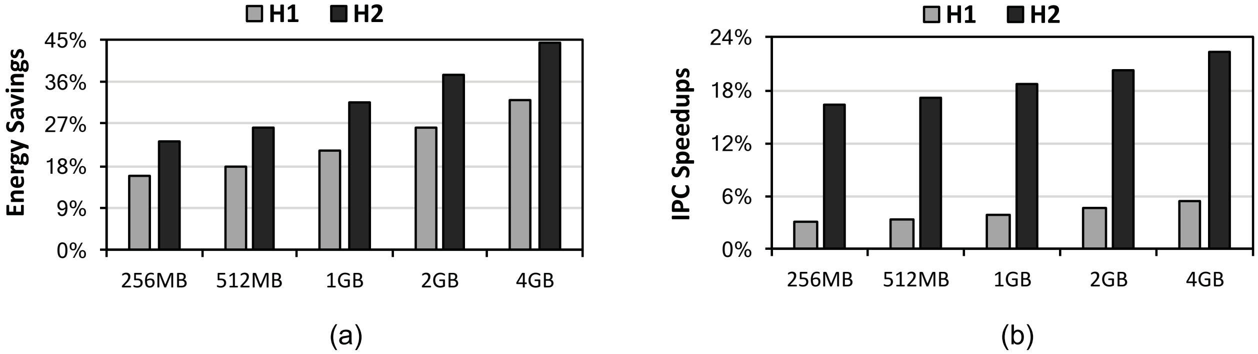
6.1 Energy Analysis

Static Energy: As shown in the Figure 6, the static energy savings of our proposed H1 architecture is about 35.4% on average compared with the baseline DRAM cache. The reason behind the savings is that STT-RAM with small leakage can effectively reduce the static energy. Similarly, the proposed H2 architecture results in 16.4% static energy reduction shown in the Figure 7. This is because STT-RAM is relaxed to reduce write energy by incurring some leakage power increment, but it is still smaller than the leakage power of DRAM.

Dynamic Energy: As presented in the Figure 6, We also observe there is average 17.6% dynamic energy reduction in H1 architecture compared with the baseline. Even for the write-intensive benchmark (e.g. *hmmer* and *perlbench*), there is average 6.5% dynamic energy saving. Because the DRAM region of our hybrid cache is configured large enough to buffer write-intensive request without accessing STT-RAM region, and the STT-RAM eliminates significant refresh energy compared with DRAM. However, there are negative energy savings for the benchmark with large datasets (e.g. mcf and *gcc*), where the input data can be the 0.5x∼1x size of DRAM cache, hence STT-RAM region needs to be frequently accessed incurring high write energy. Considering the result of H2 architecture in the Figure 7, the dynamic energy saving is more balanced in the different benchmarks with 35.5% on average. Because STT-RAM is optimized to have small write energy, and also we select an optimal retention time to balance the refresh energy and write energy.

Total Energy: The total cache energy is reduced by 21.2% and 31.6% in H1 and H2 respectively, compared to the DRAM cache baseline. This energy saving can be attributed to the following reasons: 1) the static





(a)

(b)

Figure 9: Sensitivity analysis of different hybrid cache size.

energy occupies up to 40% of the total cache energy; 2) the volatile STT-RAM with an optimal retention time reduces both the static energy and dynamic energy.

6.2 Performance Analysis

Figure 8 shows the IPC speedups of the proposed hybrid caches compared with the baseline. For H1 architecture, the performance is improved by 4.4% on average. The performance speedup comes from the proposed tag management policy that avoids the STT-RAM tag long-latency access. However, there is some negative improvement for the benchmarks with large input, which results from 1) frequently writing into conventional STT-RAM with long latency; 2) DRAM cache miss when the input datasets are larger than the DRAM cache capacity; 3) 64B random data access in the 4KB memory page caused by the DRAM cache miss.

For H2 architecture, there are 18.8% performance speedups, and we notice the results are more balanced than the H1. This is because STT-RAM is optimized to have comparable latency to the DRAM. Further, the proposed Tags-in-STT-RAM policy replaces DRAM tag access with the low-latency STT-RAM tag. Also, it is noticed that PARSEC benchmarks have more performance improvement than SPEC benchmarks. This is because the parallel characteristics of PARSEC benchmarks can meet more processing demand of CMP systems.

6.3 Sensitivity Analysis

To better understand the effect of hybrid cache capacity on the energy and performance for the HPC systems, we change the capacity to 256MB, 512MB, 1GB, 2GB and 4GB respectively. As Figure 9(a) shows, we observe the total cache energy saving increases with the larger cache capacity. This is because: (1) larger caches have more static energy consumption, which can be effectively removed by the non-volatile STT-RAM; (2) hybrid cache with larger capacity can buffer more data written in the DRAM region, which can reduce write operation in the STT-RAM compared with small cache capacity; (3) dynamic energy is proportional to the size of the cache, and the larger volatile STT-RAM has smaller dynamic energy due to the optimal retention time in our design.

As shown in the Figure 9(b), the performance also increases with larger cache capacity. It can be attributed to the following reasons: 1) the DRAM cache with larger capacity has higher hit rate, which can reduce off-chip memory access; 2) larger capacity increases hit rate in the low-latency region of hybrid cache, e.g. the DRAM region in the H1 architecture and STT-RAM region in the H2 architecture. However, it is noticed that the average tag access time depends on the cache storage size, and larger tag array requires more time for tag access and comparison, which may also impact the performance.

7 CONCLUSIONS

In this paper, we identify that DRAM cache can be used in the future HPC systems to improve performance, but it has the disadvantage of high power consumption. Thus, we present hybrid cache design for the future HPC systems to improve both energy and performance efficiency. First, we observe the DRAM cache with a large capacity has high leakage power, and the large hybrid cache using non-volatile STT-RAM is proposed to reduce static energy. Second, we propose to use volatile STT-RAM as a part of hybrid cache to reduce both dynamic and static energy of the DRAM cache. Finally, we propose tag management policy based on our hybrid cache to improve performance. The results show that energy is reduced by 31.6% and performance is improved by 18.8% on average.

REFERENCES

Bienia, C., S. Kumar, J. P. Singh, and K. Li. 2008. “The PARSEC benchmark suite: characterization and architectural implications”. In *Proceedings of PACT*.

Binkert, N., B. Beckmann, G. Black, S. K. Reinhardt, J. Saidi, D. R. Hower, T. Krishna, S. Sardashti et al. 2011. “The gem5 simulator”. *ACM SIGARCH Computer Architecture News*.

Cong, J., G. Gururaj, and Y. Zou. 2011. “An energy-efficient adaptive hybrid cache”. In *Proceedings of ISLPED*.

Das, S., T. M. Aamodt, and W. J. Dally. 2015. “SLIP: reducing wire energy in the memory hierarchy”. In *Proceedings of ISCA*.

Diao, Z., Z. Li, S. Wang, Y. Ding, A. Panchula, E. Chen, L.-C. Wang, and Y. Huai. 2007. “Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory”. *Journal of Physics: Condensed Matter*.

He, J., and J. Callenes-Sloan. 2016. “Reducing the energy of a large hybrid cache”. In *Proceedings of ICECS*.

Henning, J. L. 2006. “SPEC CPU2006 benchmark descriptions”. *ACM SIGARCH Computer Architecture News*.

Huang, C.-C., and V. Nagarajan. 2014. “ATCache: reducing DRAM cache latency via a small SRAM tag cache”. In *Proceedings of PACT*.

Jevdjic, D., G. H. Loh, C. Kaynak, and B. Falsafi. 2014. "Unison cache: A scalable and effective die-stacked DRAM cache". In *Proceedings of MICRO*.

Jevdjic, D., S. Volos, and B. Falsafi. 2013. "Die-stacked DRAM caches for servers: hit ratio, latency, or bandwidth? have it all with footprint cache". *Proceedings of ISCA*.

Kultursay, E., M. Kandemir, A. Sivasubramaniam, and O. Mutlu. 2013. "Evaluating STT-RAM as an energy-efficient main memory alternative". In *Proceedings of ISPASS*.

Li, J., C. J. Xue, and Y. Xu. 2011. "STT-RAM based energy-efficiency hybrid cache for CMPs". In *Proceedings of VLSI-SoC*.

Li, S., J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. 2009. "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures". In *Proceedings of MICRO*.

Li, T., Y. Ren, D. Yu, S. Jin, and T. Robertazzi. 2013. "Characterization of input/output bandwidth performance models in NUMA architecture for data intensive applications". In *Proceedings of ICPP*.

Liu, J., B. Jaiyen, R. Veras, and O. Mutlu. 2012. "RAIDR: Retention-aware intelligent DRAM refresh". In *Proceedings of ISCA*.

Loh, G. H., and M. D. Hill. 2011. "Efficiently enabling conventional block sizes for very large die-stacked DRAM caches". In *Proceedings of MICRO*.

Poremba, M., S. Mittal, D. Li, J. S. Vetter, and Y. Xie. 2015. "DESTINY: A Tool for Modeling Emerging 3D NVM and eDRAM caches". In *Proceedings of DATE*.

Smullen, C. W., V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan. 2011. "Relaxing non-volatility for fast and energy-efficient STT-RAM caches". In *Proceedings of HPCA*, pp. 50–61.

Sun, G., X. Dong, Y. Xie, J. Li, and Y. Chen. 2009. "A novel architecture of the 3D stacked MRAM L2 cache for CMPs". In *Proceedings of HPCA*.

Wang, Z., D. A. Jiménez, C. Xu, G. Sun, and Y. Xie. 2014. "Adaptive placement and migration policy for an STT-RAM-based hybrid cache". In *Proceedings of HPCA*.

Wu, X., J. Li, L. Zhang, R. Speight, and Y. Xie. 2009. "Hybrid cache architecture with disparate memory technologies". In *Proceedings of ISCA*.

Zhao, L., R. Iyer, R. Illikkal, and D. Newell. 2007. "Exploring DRAM cache architectures for CMP server platforms". In *Proceedings of ICCD*.

AUTHOR BIOGRAPHIES

JIACONG HE received M.S. degree in Computer Engineering from the Illinois Institute of Technology. He currently is a Ph.D. student in the Electrical Engineering Department at the University of Texas at Dallas. His research interests include memory systems, high performance computing and dependable systems. His email address is jiacong.he@utdallas.edu.

JOSEPH CALLENES-SLOAN is an Assistant Professor in the Erik Jonsson School of Engineering & Computer Science at the University of Texas at Dallas. He holds a Ph.D. in Electrical and Computer Engineering at the University of Illinois in Urbana-Champaign. His research interests include computer architecture, scientific and high performance computing, low power design and fault tolerance. His email address is jcallenes.sloan@utdallas.edu.