IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 21, 2020, accepted February 2, 2020, date of publication February 7, 2020, date of current version February 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972305

Statistical Behavior Guided Block Allocation in Hybrid Cache-Based Edge Computing for Cyber-Physical-Social Systems

FANFAN SHEN[1], CHAO XU[1], AND JUN ZHANG[2]

[1]School of Information Engineering, Nanjing Audit University, Nanjing 211815, China
[2]School of software, East China University of Technology, Nanchang 330013, China

Corresponding author: Chao Xu (xuchao@nau.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61902189, Grant 71972102, and Grant 61662002, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180821, in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant 18KJB520026, in part by the Collaborative Innovation Center of Audit Information Engineering and Technology under Grant 18CICA11, in part by the MOE Project of Humanities and Social Sciences under Grant 19YJAZH100, and in part by the Project of Jiangxi Engineering Laboratory on Radioactive Geoscience and Big Data Technology under Grant JELRGBDT201905.

ABSTRACT In Cyber-Physical-Social Systems (CPSS), large-scale data are continually generated from edge computing devices in our daily lives. These heterogeneous data collected from CPSS are urgently needed to be processed efficiently with low power consumption. Hybrid cache based edge computing can accelerate the computing speed for the edge devices. Hybrid cache consisting of spin-transfer torque RAM (STT-RAM) and static RAM (SRAM) has been proposed as last level cache (LLC) for energy efficiency recently in CPSS. However, the write operations on STT-RAM suffer from considerably higher energy consumption as well as longer latency than SRAM, the proper allocation of data blocks has a significant effect on both energy consumption and performance in the hybrid cache. So it is very useful to adjust the data allocation for the asymmetric-access in hybrid cache. To enhance the performance of hybrid cache, this paper proposes a novel statistical behavior guided block allocation (SBOA) scheme to process CPSS data. The key idea is to estimate the cache block characteristics based on the statistical behavior of data read/write re-references. We design a theoretical analysis model to optimize the energy consumption and guide block allocation in both SRAM region and STT-RAM region. Experimental results demonstrate that the proposed scheme reduces the dynamic energy consumption by 18.5%, and reduces execution time by 7.4% on average compared to the baseline with negligible overhead.

INDEX TERMS Statistical behavior, block allocation, non-volatile memory, hybrid cache.

I. INTRODUCTION

As the rapid development of the Internet of Things (IoT), the cyber, physical, and social worlds are integrated together. It is also referred as Cyber-Physical-Social Systems (CPSS) [1]–[3]. In CPSS, large-scale data are generated everyday from the edge computing devices, thus edge computing methods become more attractive and can improve performance for edge devices. The demand of edge computing with high efficiency and low power consumption are increasing in our living environments [4]–[6].

At the same time, the Moore's Law continues, more and more processing cores are integrated into a single chip [7]. Thus, the demand for large last level cache (LLC) has become increasingly important to mitigate the memory wall problem. Traditional SRAM-based LLC encounters many challenges such as the high leakage power and poor scalability [8]–[10]. To address these issues, various emerging non-volatile memory technologies have been extensively studied recently. In particular, spin-transfer torque RAM (STT-RAM) is gaining attention as a promising candidate for building caches in the future in CPSS. Compared to SRAM, STT-RAM has attractive features including better scalability, lower leakage power and higher storage density [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Piccialli.

Despite of these advantages, the major obstacles to use STT-RAM as on-chip caches are long write latency and high write energy. To utilize the benefit of both SRAM and STT-RAM, the hybrid cache architectures have been explored to optimize the inefficient write operations. Many architectural techniques have been proposed to design way-based [11], region-based [12] and different levels hybrid caches [13]. Based on these designs, a set of block placement and migration policies are used to map write-intensive blocks from STT-RAM to SRAM [14]–[21]. Since the cost of write operations with SRAM is much smaller than STT-RAM, these polices help to reduce the number of write operations in STT-RAM and thereby improving the write performance as well as reducing the write energy.

Previous works [14]–[22] improve the cache efficiency from various perspectives. However, it is not sufficient to identify write-intensive blocks only based on the access types (core, prefetch and demand-write [14], read/write [15]). Some approaches incur frequent block migrations which cause migration overheads [12], [16], [17]. Some compilation techniques require the compiler to provide static hints [18]–[20], which are impractical in some cases. Recent work proposes a trace-based prediction hybrid cache to predict write burst blocks dynamically [21], but this design brings significant overhead that can not be ignored. Kim *et al.* [22] proposed a hybrid cache architecture based on reuse distance prediction, it is the distance from the first reference to the last reference. However, this approach focuses on exclusive hybrid cache, which limits its extensive use.

Considering these issues, we observe that the cache access behavior could be quantified with statistical data. In this work, we propose a novel approach called Statistical Behavior guided blOck Allocation (SBOA) scheme to process CPSS data. SBOA classifies the cache blocks into three types: read-only, write-only and interleaved-access. Then we focus on the energy optimization of interleaved-access blocks. SBOA makes the block allocation decision based on the read/write statistical information gathered from the historical data. The evaluation results show that SBOA achieves the dynamic energy reduction by 18.5% and performance improvement by 7.4% on average compared with the baseline. In addition, compared with the adaptive block placement and migration policy (APM) [14], SBOA simultaneously reduces the dynamic energy consumption by 6.4% and execution time by 3.3% on average respectively. Specifically, the main contributions of this paper are summarized as follows:

- We provide theoretical analysis of the energy consumption in hybrid cache with data allocation.
- We propose a statistics-based SBOA scheme to improve hybrid cache efficiency in CPSS.
- We evaluate the effectiveness of the proposed technique and the experiments show that it improves the energy efficiency as well as performance.

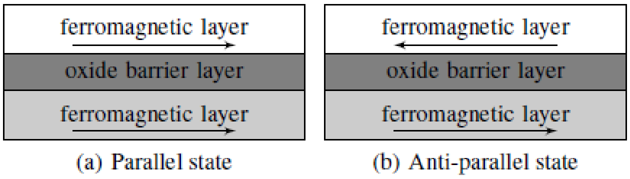


FIGURE 1. Magnetic tunnel junction structure of STT-RAM.

The rest of this paper is organized as follows. The background and preliminary study are introduced in Section II. Section III presents the proposed approach. Section IV describes the evaluation methodology and experimental results. Section V discusses the related work. Finally, Section VI concludes this paper.

II. BACKGROUND AND PRELIMINARY STUDY

This section first introduces the fundamentals of STT-RAM and hybrid cache, then we present the preliminary study of this work.

A. STT-RAM AND HYBRID CACHE OVERVIEW

The basic storage element in STT-RAM is magnetic tunnel junction (MTJ). Each MTJ has two ferromagnetic layers separated by one oxide barrier layer. The resistance of each MTJ depends on the relative magnetization directions of the two ferromagnetic layers, thereby creating two possible states: parallel and anti-parallel states, corresponding two resistance states are shown in Fig. 1. A low current is enough to read the MTJ state, while a high current is required to change the magnetic state, which involves long write latency and high write energy consumption [8], [17].

Compared to SRAM, STT-RAM consumes much lower leakage power due to its non-volatility. In order to combine the benefits of these two memory technologies, researchers have proposed to construct a hybrid cache consisting of both SRAM and STT-RAM [11]–[13]. Each cache set has a large region of STT-RAM cache blocks and a small region of SRAM cache blocks. Since the static energy of STT-RAM is very minimal and our target is to optimize the dynamic energy. So we do not consider the static energy consumption in STT-RAM. The hybrid cache architecture relies on an intelligent block allocation policy to bridge the performance and power gaps between STT-RAM and SRAM [14].

B. PRELIMINARY STUDY

In the preliminary study, we investigate the read and write operation behaviors of several multi-threaded workloads chosen from the PARSEC benchmark suite [23]. They are executed on gem5 simulator [24] with a two level cache architecture. The detailed configuration can be found in Section IV-A. A group of selected applications are shown in Table 1. We collect the statistical data of read-only, write-only and interleaved-access blocks. The data are obtained when the cache blocks are evicted from the LLC. As can

TABLE 1. Reading and writing statistics of cache blocks in LLC.

Benchmarks	Read-only	Write-only	Interleaved-access
canneal	26.2%	19.7%	54.1%
dedup	29.3%	22.9%	47.8%
ferret	7.8%	9.7%	82.5%
x264	21.4%	17.6%	61%

be seen from the table, different applications have differing read and write behaviors. The interleaved-access blocks dominate the cache accesses. Especially for *ferret* application, the number of interleaved-access blocks are up to 82.5%, which shows more potential for block allocation in the hybrid cache.

Based on the observation above, these access behaviors show the potential benefits for energy reduction. Perfect allocation of all cache blocks with absolutely right prediction could minimize the energy consumption. To this end, the write-only blocks should be placed in SRAM due to the high write energy of STT-RAM, while the read-only blocks should be placed in STT-RAM. Reads and writes can be distinguished to achieve better power and performance. More importantly, the majority of interleaved-access blocks should be carefully addressed, especially for read-intensive and write-intensive cache blocks. This motivates us to explore a suitable strategy that can effectively place the interleaved-access blocks in the SRAM region or STT-RAM region.

III. STATISTICAL BEHAVIOR GUIDED BLOCK ALLOCATION

In this section, we present the technical details of the proposed approach. The problem definition is introduced first. Then, we show the detailed architecture of SBOA. Finally, we theoretically analyze the benefit of the block allocation policy for energy consumption based on statistical behavior.

A. PROBLEM DEFINITION

The objective of this work is to reduce the cache access energy of hybrid cache by intelligent allocating cache blocks appropriately. Since the static energy consumption is the basic characteristic of the peripheral circuits, so no matter where the cache block is stored, the static energy consumption is the same compare to the baseline. Furthermore, the static energy is very minimal for STT-RAM based hybrid cache and our target is to optimize the dynamic energy. So we do not consider the static energy consumption. To achieve this goal, we consider both read and write operations of each blocks in the hybrid cache. Initially, if a block A is loaded into SRAM of hybrid cache, the total access energy caused by this block can be calculated in Equation (1). The notations and definitions of read and write operations to hybrid LLC used in this study are listed in Table 2.

TABLE 2. Definitions of the target problem.

Notation	Definition
N_r	The number of reads to the block until its eviction
N_w	The number of writes to the block until its eviction
E_r^{STT}	Read energy of the STT-RAM region in LLC
E_w^{STT}	Write energy of the STT-RAM region in LLC
E_r^S	Read energy of the SRAM region in LLC
E_w^S	Write energy of the SRAM region in LLC
P_r^i	Probability of $N_r = i$
P_w^j	Probability of $N_w = j$

$$E_{SRAM} = N_r \times E_r^S + N_w \times E_w^S + E_w^S \tag{1}$$

It means that, since this block is loaded into SRAM, it has been read N_r times and written N_w times until its eviction. E_w^S represents the write energy of the initial allocation. Similarly, if the block A is loaded into STT-RAM of hybrid cache, the total access energy caused by this block will be changed to that in Equation (2).

$$E_{STT} = N_r \times E_r^{STT} + N_w \times E_w^{STT} + E_w^{STT} \tag{2}$$

Obviously, we can reduce the access energy by allocating this block in STT-RAM when we have $E_{STT} < E_{SRAM}$, and vice versa. Therefore, we can obtain the condition of the block placement in Equation (3). This indicates that the energy cost of this block is relatively low if the number of reads and writes satisfy this condition. Otherwise, the block is placed into the SRAM region. According to this analysis, the block allocation policy has demonstrated the great potential in energy reduction.

$$\frac{N_r}{N_w + 1} > \frac{E_w^{STT} - E_w^S}{E_r^S - E_r^{STT}} \tag{3}$$

Note that, it is impossible to know exactly the N_r and N_w values of all cache blocks when they are first introduced into the LLC. So can we predict the read and write numbers of cache blocks ahead of time? Of course, it is possible to get the approximate estimation of the block according to the statistical data. Then we can use these informations to guide the block allocation.

B. SBOA ARCHITECTURE

As shown in Fig. 2, SBOA is implemented with the sampled sets and predictor table, which are constructed with SRAM due to their small size. The sampled sets are in a separate hardware called sampler. Each block in the sampler contains a valid bit, a 16-bit tag, a 13-bit PC, and an 8-bit read/write times field [21], [22]. The sampler keeps track of the cache access behavior for each block with the instruction address that has accessed the block. The instruction is based on program counter (PC) [25]. To reduce the area and energy overhead, we choose $\frac{1}{32}$ of the entire sets as the sampled sets. The predictor table records the prediction information with a flag bit and a 13-bit PC for indexing. It has 8192-entry. This table is used to predict whether the next access with the given PC will be placed in SRAM (flag=0) or STT-RAM (flag=1).

Fig. 3 presents the overall flow diagram of SBOA architecture. For each access to a sampled set in the LLC, the corresponding PC, read/write field in the sampler are updated. The read/write field is initialized to zero on block insertion.

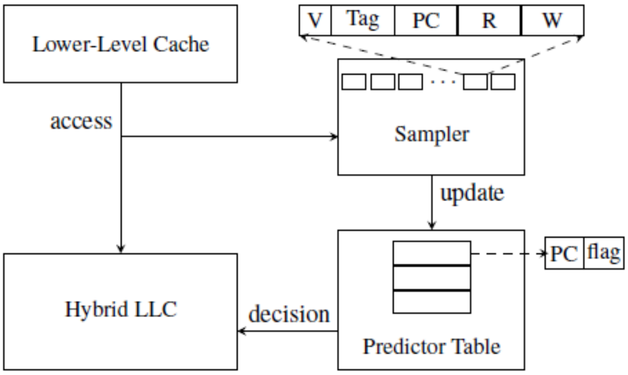


FIGURE 2. SBOA Architecture. Cache line in the sampler including V: Valid bit, Tag: Patial tag, PC: Program counter, R: Read times and W: Write times.

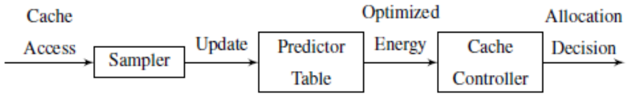


FIGURE 3. Overall flow diagram of SBOA.

When a block is evicted from the sampler, we gather the statistical information with read and write times. Subsequently, the belonging of this block is calculated and the prediction flag bit is updated in the predictor table. The detailed update policy of this table will be discussed in the next subsection III-C.

According to the flags with optimized energy in the predictor table, the cache controller decides the block allocation when the LLC receives a block insertion request. It will check the flag that is indexed by the PC from the request. If the flag is 0, the block is loaded into SRAM. Otherwise, it is loaded into the STT-RAM.

Note that we do not need to migrate blocks between the SRAM region and the STT-RAM region. This is because the migration overhead is not negligible in terms of both area and energy. If the prediction of the block allocation is accurate enough, there are only a few blocks that need to migrate. Nonetheless, the energy consumption can be reduced further with the block migration policy. To demonstrate the value of accurate block allocation, we do not consider block migration throughout this paper.

C. ENERGY OPTIMIZATION OF SBOA

In order to guide the block allocation, it is very important to know the cache block access behavior when they are first inserted into the hybrid cache. For example, if the accesses to a cache block are dominated by the write operations in the future, allocating this cache block in the SRAM region can significantly reduce energy consumption. On the contrary, this block is suitable for allocating in the STT-RAM region.

Fig. 4 shows the distribution of the cache access characteristics from the statistical results. The detailed configuration is introduced in Section IV-A. We collect the access characteristics for all cache blocks and then classify them into three types:

a) Read-only: If all the accesses to a cache block are read operations until the bock is evicted, we regard this block as read-only. It means $N_r > 0$ and $N_w = 0$.

b) Write-only: If all the accesses to a cache block are write operations until the bock is evicted, we regard this block as write-only. It means $N_w > 0$ and $N_r = 0$.

c) Interleaved-access: if the accesses to a cache block are interleaved with read and write operations, we regard this block as interleaved-access. It means $N_r > 0$ and $N_w > 0$.

As can be seen from the Fig. 4, the applications have various distributions. On average(average mean, AMEAN), there are 25.8% read-only cache blocks, 14.9% write-only cache blocks and 59.3% interleaved-access cache blocks. The majority of the cache access behaviors are interleaved access, which shows the potential for the energy optimization.

We first identify the read-only or write-only block by tracking the block accesses until the block is evicted in the sampled sets. If N_r (N_w) is 0 when the block is evicted, the block is regarded as write-only (read-only) and then the prediction flag is updated to 0 (1). The predicted read-only blocks will be allocated in the STT-RAM region while the write-only blocks will be allocated in the SRAM region in the future. Thus, the energy consumption is reduced.

For the interleaved-access block, we introduce the definition of N_r and N_w probability to get the approximate estimation. The probability is obtained from the statistical behavior of the cache blocks in the sampled sets. Let $X_{N_r=i}$ denotes the number of data that have N_r equals to i in the sampler. Then a N_r probability P_r^i is calculated in Equation (4).

$$P_r^i = \frac{X_{N_r=i}}{\sum_{k=0}^{\infty} X_{N_r=k}} \tag{4}$$

Similarly, Let $Y_{N_w=j}$ denotes the number of data that have N_w equals to j in the sampler. Then a N_w probability P_w^j is calculated in Equation (5).

$$P_w^j = \frac{Y_{N_w=j}}{\sum_{k=0}^{\infty} Y_{N_w=k}} \tag{5}$$

To make it more clearly, let (N_r, N_w) denotes a pair of read/write times of a cache block. Supposing that we have (1, 2), (2, 3), (2, 3), (4, 3) and (5, 1). Then the P_r^2 is $\frac{2}{5}$ and the P_w^3 is $\frac{3}{5}$.

Considering the statistical behavior of the cache block in the sampler, if a block A is likely to be allocated in SRAM (STT-RAM), the energy consumption is noted as E_{SRAM}^A (E_{STT}^A). These two values are described in Equation (6) and (7).

$$E_{SRAM}^A = P_r^i \times i \times E_r^S + P_w^j \times j \times E_w^S + E_w^S \tag{6}$$

$$E_{STT}^A = P_r^i \times i \times E_r^{STT} + P_w^j \times j \times E_w^{STT} + E_w^{STT} \tag{7}$$

The energy consumption is reduced if the block A is placed in STT-RAM in the future when $E_{STT}^A < E_{SRAM}^A$. After comparing these two equations, we derive the condition to trigger the block allocation in STT-RAM by statistical behavior,

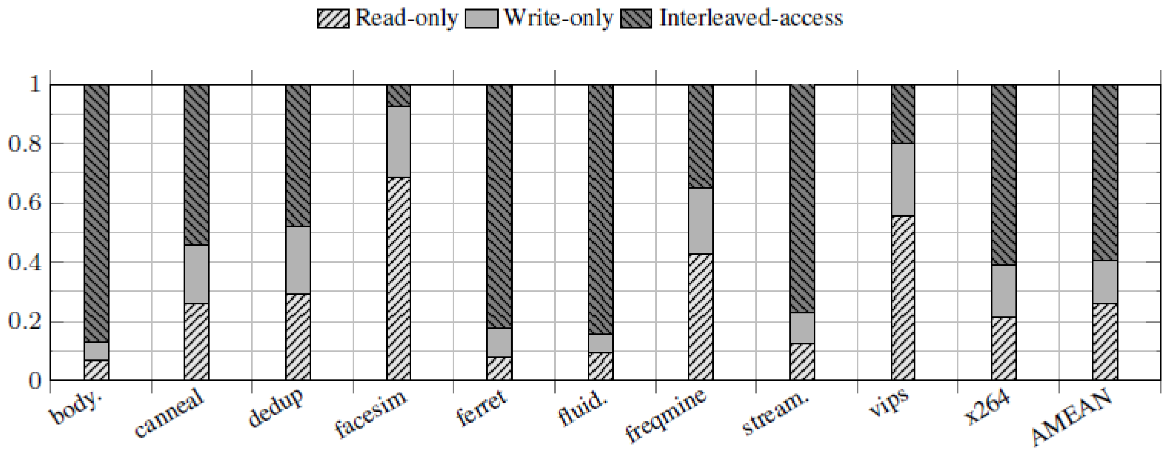


FIGURE 4. Distribution of the cache access characteristics.

as shown in Equation (8). Subsequently, the prediction flag of this block *A* is updated to 1 and stored in the predictor table if the condition is satisfied. On the contrary, the prediction flag is updated to 0.

$$\frac{P_r^i \times i}{P_w^j \times j + 1} > \frac{E_w^{STT} - E_w^S}{E_r^S - E_r^{STT}} \tag{8}$$

Note that it is easy to get the left-hand value in Equation (8) when the interleaved-access block is evicted in the sampled sets. For example, supposing that we have five cache blocks in the sampled sets and the corresponding read/write times are (1, 2), (2, 3), (2, 3), (4, 3) and (5, 1). When the second block is evicted from the sampled sets, we can calculate the left-hand value as follows:

$$\frac{P_r^i \times i}{P_w^j \times j + 1} = \frac{P_r^2 \times 2}{P_w^3 \times 3 + 1} = \frac{\frac{2}{5} \times 2}{\frac{3}{5} \times 3 + 1} \approx 0.29 \tag{9}$$

For the right-hand value in Equation (8), we can calculate it in Equation (10). The read/write energy in hybrid cache are listed in Table 3.

$$\frac{E_w^{STT} - E_w^S}{E_r^S - E_r^{STT}} = \frac{0.685 - 0.308}{0.308 - 0.216} \approx 4.1 \tag{10}$$

With the comparison of Equation (9) and (10), we can learn that this kind of block should be allocated in SRAM. Then the PC and flag of this block are updated in the predictor table. In the calculation process, we need to obtain read/write pairs of each blocks in the sampled sets. Fortunately, we have recorded these informations in the sampled sets as shown in Figure 2.

Obviously, where to place the block is based on the benefits of the energy reduction with the value of statistics P_r^i and P_w^j. This is the reason why we call our technique as a statistics based block allocation approach (SBOA).

IV. EVALUATION

In this section, we evaluate the effectiveness of the proposed statistical behavior guided block allocation (SBOA) scheme. The evaluation methodology is introduced first, and then the corresponding experimental results, including prediction accuracy, energy consumption, execution time and overhead, are summarized with extensive analysis.

TABLE 3. Simulation parameters.

Parameters	Value
Processor	4 cores, 2GHz
L1 cache	Private, 32KB, 2-way, 64B, LRU, 2 cycles
Hybrid LLC	Shared, 64B, LRU, 8MB, 16-way
	4-way for SRAM
	12-way for STT-RAM
	SRAM access latency: 25 cycles
	SRAM access energy: 0.308nJ
	STT-RAM R/W latency: 25/60 cycles
	STT-RAM R/W energy: 0.216/0.685nJ
Memory	4GB, 1600MHz, 12.8GB/s, 200 cycles

A. METHODOLOGY

We implement SBOA in a popular full-system simulator gem5 [24]. Table 3 shows the parameters of the baseline configuration. It is configured to model a four-core and two levels of caches. The classic memory model in the gem5 is modified to implement the way-based hybrid last-level cache. The cache parameters are obtained from a modified CACTI [26] and NVSim [27].

We select a set of PARSEC benchmarks [23] with *simlarge* input sets to evaluate the proposed scheme. They have different intensity of read/write operations. To obtain a reasonable evaluation, all benchmarks are fast forwarded to the region of interest (ROI) and then we run the benchmarks for a maximum of two billion instructions.

To implement the proposed SBOA scheme, we add the sampled sets and prediction table in the LLC. The allocating block function in the cache implementation module is modified to decide where to place a cache block when the block is inserted into the LLC. There is no initial configuration of the predictor. We can get the read/write statistics from the sampler after the initial running of the benchmark, and then compute the energy and probability.

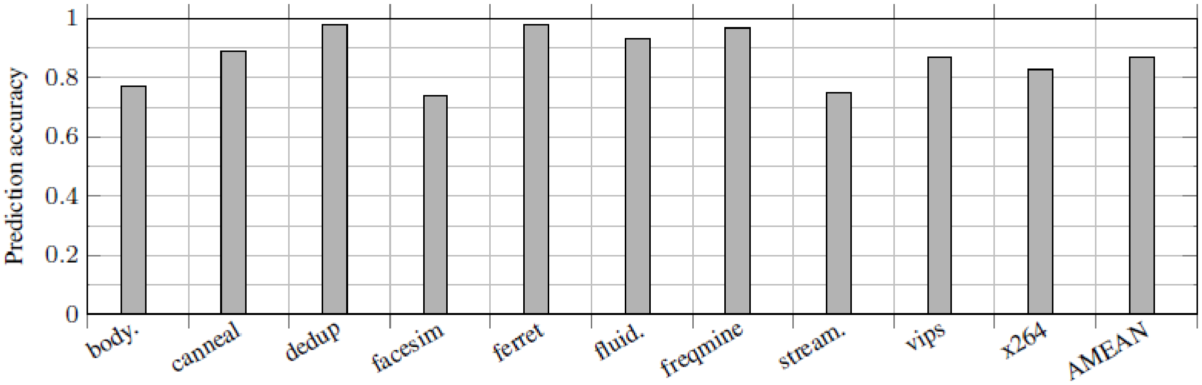


FIGURE 5. Prediction accuracy of SBOA.

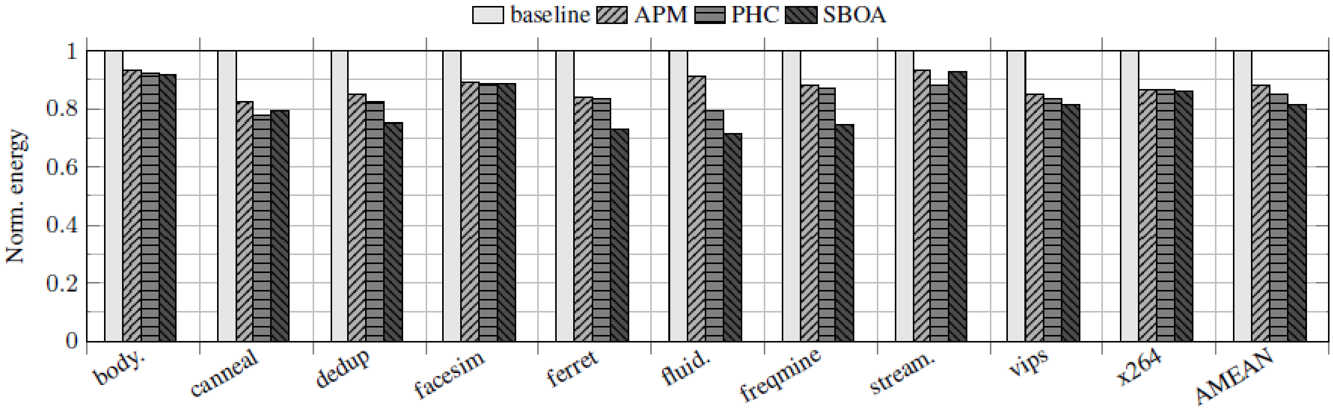


FIGURE 6. Normalized energy consumption.

To evaluate the effectiveness of the proposed scheme, the baseline scheme is the cache accesses without optimization. A modified version of adaptive placement and migration (APM) [14] scheme and prediction hybrid cache (PHC) scheme [21] are selected to compare with SBOA. In APM strategy, the block placement is determined by the access pattern of core/prefetch/demand-write. In PHC strategy, the blocks are identified with hot trigger instructions.

B. PREDICTION ACCURACY

The prediction accuracy is very important for SBOA. The energy reduction and performance improvement are maximization when all blocks are allocated correctly. However, it is not practical for all right prediction. Any mis-prediction will introduce extra energy consumption and performance penalty.

In order to obtain the prediction accuracy, we keep track of the predicted block in the LLC with a trace bit (This bit is not needed in the real environment). When the predicted block is evicted in the LLC, the actual allocation of this block is calculated again. If it is equal to the trace bit, this is a correct prediction. Otherwise, it is a mis-prediction.

As shown in Fig. 5, SBOA achieves a high prediction accuracy by 87.1% on average. The accuracy varies among different applications. For *dedup* workload, the prediction accuracy reaches up to 98.2% compared with the baseline. This indicates that the statistical behavior can reflect the characteristics of the cache blocks with the actual enhancement of accuracy. The majority of the cache blocks can be allocated in the appropriate region for the energy reduction. This is the reason why the cache efficiency is improved.

C. ENERGY CONSUMPTION

Fig. 6 shows the dynamic energy consumption of the proposed scheme over the baseline scheme. The baseline scheme is the cache accesses without optimization. As expected, SBOA outperforms the baseline for most of the evaluated PARSEC workloads. The energy reduction is related to the prediction accuracy generally. With high prediction accuracy, SBOA has more potential to place the predicted blocks in the right region. On the contrary, the energy reduction is relatively low. For example, the *dedup* workload reduces the energy consumption by about 25.3%, while the *streamcluster* workload reduces energy consumption by 7.5% with low prediction accuracy compared with the baseline.

In summary, SBOA reduces the dynamic energy consumption of LLC by 18.5%, 6.4% and 3.5% on average compared to the baseline, APM and PHC, respectively. This can be

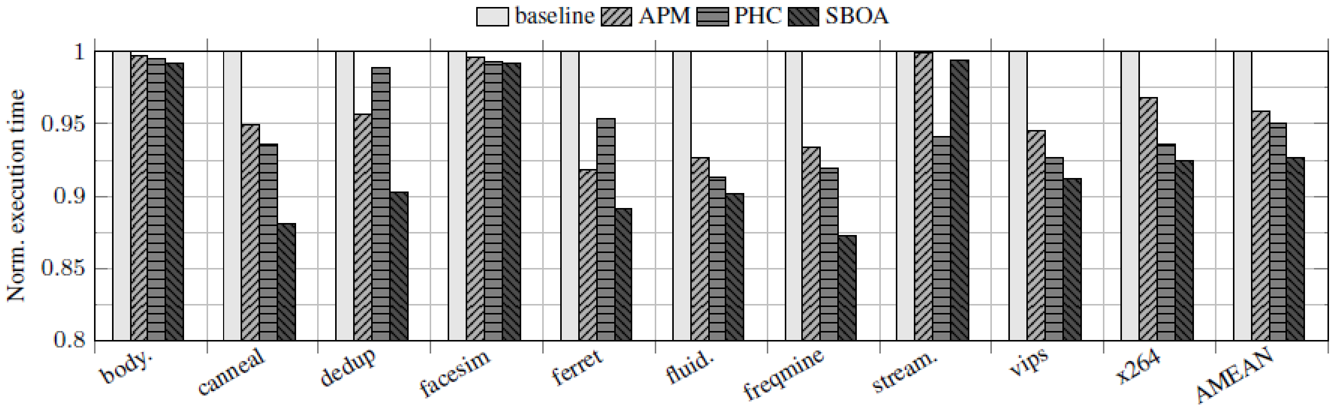


FIGURE 7. Normalized energy consumption.

explained by our scheme identifies the characteristic of the cache blocks efficiently than APM and PHC, and thus, more cache blocks are handled in the right region of the hybrid cache. Therefore, allocating the cache blocks intelligently is beneficial.

Furthermore, the reason for the energy reduction is that SBOA obeys the energy optimization goal to find out the most potential region for the blocks according to the statistical behavior of the cache blocks. However, APM identifies the block access pattern only by write access types. This will miss some write-intensive blocks and thereby causing less energy reduction.

D. EXECUTION TIME

As depicted in Fig. 7, the execution time of the workloads is normalized to the baseline. For most of the workloads, our scheme outperforms the baseline, APM and PHC. Specifically, the results of performance improvement are similar to that of the energy reduction, but less significant. This is because our goal is energy optimization. The total execution time is reduced by 7.4%, 4.1% and 4.9% respectively compared with the baseline, APM and PHC. For the *freqmine* workload, the performance is improved up to 12.7% compared to the baseline. The *dedup* workload achieves performance improvement up to 9.7%. We can see that the performance improvement is also related to the prediction accuracy. Because the overhead of mis-prediction is reduced and thereby maximizing the benefits of block allocation. It is interesting that the high prediction accuracy promotes both energy reduction and performance improvement.

What's more, the major contribution of this performance improvement comes from the reduction of the write operations to the STT-RAM region, each of which takes two times longer than that in the SRAM region. This enhancement is induced by the energy optimization goal with intelligent block allocation.

E. OVERHEAD OF THE PROPOSED SCHEME

The additional storage overhead introduced by the proposed scheme contains the sampled sets and predictor table. A valid bit, a 16-bit tag, a 13-bit PC and 8-bit read/write times counters are added to each block in the sampled sets (256 sets out of 8192 sets in our environment). The predictor table has 1-bit prediction flag and 13 bits PC for indexing with 8192-entry. Therefore, the total storage overhead of our technique is only 37KB (23KB+14KB, 0.45%) in the LLC, which is negligible.

According to the latency and power model from CACTI [26], the sampler and predictor table are not on the critical path of the cache accesses, and thus do not increase the cache access latency and energy consumption. This is because the sampler is updated in parallel with the cache accesses and the predictor table is accessed only on the cache misses. Therefore, the runtime overheads are also negligible.

The experimental methodology is tested on the baseline configuration. The proposed scheme does not need to set the parameter, so the effect of the proposed schemes is very minimal among different configurations.

V. RELATED WORK

In this section, we briefly overview the related work about hybrid cache.

Many researchers have explored the hybrid cache architecture with various memory technologies. In most cases, it is a combination of STT-RAM and SRAM [11]–[13], [28]. Chen *et al.* [11] proposed a reconfigurable hybrid cache architecture by powering on/off SRAM/NVM arrays in a way-based manner. Wu *et al.* [12] proposed inter cache level and intra cache level hybrid cache architecture with disparate memory technologies. Zhao *et al.* [13] proposed a bandwidth-aware reconfigurable hybrid cache. These works are addressed in the architectural level to improve the hybrid cache efficiency, while our work is orthogonal to those approaches for the same goal.

In order to fully utilize the benefit of hybrid cache, researchers mainly focus on the cache block allocation policy that allocating the blocks to the SRAM region or STT-RAM region on demand [14], [15], [29] and the cache block migration policy to move write-intensive blocks to the SRAM region [16], [17]. Wang *et al.* [14] proposed a block allocation and migration policy based on the access patterns of hybrid LLC. Li *et al.* [15] proposed a novel hybrid cache architecture, and they also presented the micro-architectural mechanisms to make the hybrid cache robust to workloads with different write patterns. Jadidi *et al.* [16] proposed migrating frequently written cache blocks to SRAM and pushing rarely-written or read-only ones into STT-RAM. However, frequently migrating data in the hybrid cache incurs significant performance and energy overhead. Another researchers used the compilation techniques to optimize the block allocation or migration overhead for hybrid cache with static hints [18]–[20].

The main ideas of these works are similar to our work for that we all focus on minimizing the write operation in the STT-RAM. Even though the main idea is similar, our work has two contributions compared with prior works. First, to the best of our knowledge, this is the first attempt to utilize the theoretical analysis model for energy reduction in the hybrid cache. The other contribution is that a statistics-based scheme with historical data is proposed to guide block allocation yet prior works are fuzzy.

VI. CONCLUSION

This paper proposes a novel statistical behavior guided block allocation (SBOA) scheme to process CPSS data efficiently with low power consumption for edge computing devices. SBOA contains an energy-oriented theoretical analysis model and a low cost predictor. The model identifies the cache block characteristics from the read/write statistical behavior with historical data, which is recorded in the sampler. Then the predictor is used to guide the block allocation. Our simulation results show that the proposed technique can improve energy efficiency as well as performance with acceptable overhead in CPSS, compared to the state-of-the-art approach. The future work will focus on two directions. First, we will try to improve the energy and performance for many other hybrid caches with the theoretical analysis model. Second, we will try to reduce the storage and runtime overheads.

REFERENCES

[1] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 80–85, Nov. 2017.

[2] J. J. Zhang, F.-Y. Wang, X. Wang, G. Xiong, F. Zhu, Y. Lv, J. Hou, S. Han, Y. Yuan, Q. Lu, and Y. Lee, "Cyber-physical-social systems: The state of the art and perspectives," *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 3, pp. 829–840, Sep. 2018.

[3] M. A. Rahman, M. Y. Mukta, A. Yousuf, A. T. Asyhari, M. Z. A. Bhuiyan, and C. Y. Yaakub, "IoT Based Hybrid Green Energy Driven Highway Lighting System," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Aug. 2019, pp. 587–594.

[4] L. T. Yang, X. Wang, X. Chen, L. Wang, R. Ranjan, X. Chen, and M. J. Deen, "A multi-order distributed HOSVD with its incremental computing for big services in cyber-physical-social systems," *IEEE Trans. Big Data*, to be published.

[5] X. Wang, L. T. Yang, Y. Wang, X. Liu, Q. Zhang, and M. J. Deen, "A distributed tensor-train decomposition method for cyber-physical-social services," *ACM Trans. Cyber-Phys. Syst.*, vol. 3, no. 4, pp. 1–15, Oct. 2019.

[6] M. A. Rahman, A. T. Asyhari, S. Azad, M. M. Hasan, C. P. C. Munaiseche, and M. Krisnanda, "A cyber-enabled mission-critical system for post-flood response: Exploiting TV white space as network backhaul links," *IEEE Access*, vol. 7, pp. 100318–100331, 2019.

[7] J. Zhou, X. S. Hu, Y. Ma, J. Sun, T. Wei, and S. Hu, "Improving availability of multicore real-time systems suffering both permanent and transient faults," *IEEE Trans. Comput.*, vol. 68, no. 12, pp. 1785–1801, Dec. 2019.

[8] D. Apalkov, A. Ong, A. Driskill-Smith, M. Krounbi, A. Khvalkovskiy, S. Watts, V. Nikitin, X. Tang, D. Lottis, K. Moon, X. Luo, and E. Chen, "Spin-transfer torque magnetic random access memory (STT-MRAM)," *JETCJ. Emerg. Technol. Comput. Syst.*, vol. 9, no. 2, pp. 1–35, May 2013.

[9] J. Zhou, J. Yan, K. Cao, Y. Tan, T. Wei, M. Chen, G. Zhang, X. Chen, and S. Hu, "Thermal-aware correlated two-level scheduling of real-time tasks with reduced processor energy on heterogeneous MPSoCs," *J. Syst. Archit.*, vol. 82, pp. 1–11, Jan. 2018.

[10] J. Zhou, J. Sun, X. Zhou, T. Wei, M. Chen, S. Hu, and X. S. Hu, "Resource management for improving soft-error and lifetime reliability of real-time MPSoCs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 38, no. 12, pp. 2215–2228, Dec. 2019.

[11] Y.-T. Chen, J. Cong, H. Huang, B. Liu, C. Liu, M. Potkonjak, and G. Reinman, "Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2012, pp. 45–50.

[12] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," in *Proc. Int. Symp. Comput. Archit.*, 2009, pp. 34–45.

[13] J. Zhao, C. Xu, T. Zhang, and Y. Xie, "BACH: A bandwidth-aware hybrid cache hierarchy design with nonvolatile memories," *J. Comput. Sci. Technol.*, vol. 31, no. 1, pp. 20–35, Jan. 2016.

[14] Z. Wang, D. A. Jimenez, C. Xu, G. Sun, and Y. Xie, "Adaptive placement and migration policy for an STT-RAM-based hybrid cache," in *Proc. IEEE 20th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2014, pp. 13–24.

[15] J. Li, C. J. Xue, and Y. Xu, "STT-RAM based energy-efficiency hybrid cache for CMPs," in *Proc. IEEE/IFIP 19th Int. Conf. VLSI Syst.-Chip*, Oct. 2011, pp. 31–36.

[16] A. Jadidi, M. Arjomand, and H. Sarbazi-Azad, "High-endurance and performance-efficient design of hybrid cache architectures through adaptive line replacement," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design*, Aug. 2011, pp. 79–84.

[17] I.-C. Lin and J.-N. Chiou, "High-endurance hybrid cache design in CMP architecture with cache partitioning and access-aware policies," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 10, pp. 2149–2161, Oct. 2015.

[18] Y.-T. Chen, J. Cong, H. Huang, C. Liu, R. Prabhakar, and G. Reinman, "Static and dynamic co-optimizations for blocks mapping in hybrid caches," in *Proc. Int. Symp. Low-Power Electron. Design*, 2012, pp. 237–242.

[19] Q. Li, J. Li, L. Shi, M. Zhao, C. J. Xue, and Y. He, "Compiler-assisted STT-RAM-based hybrid cache for energy efficient embedded systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 8, pp. 1829–1840, Aug. 2014.

[20] K. Qiu, W. Zhang, X. Wu, X. Zhu, J. Wang, Y. Xu, and C. J. Xue, "Balanced loop retiming to effectively architect STT-RAM-based hybrid cache for VLIW processors," in *Proc. 31st Annu. ACM Symp. Appl. Comput.*, 2016, pp. 1710–1716.

[21] J. Ahn, S. Yoo, and K. Choi, "Prediction hybrid cache: An energy-efficient STT-RAM cache architecture," *IEEE Trans. Comput.*, vol. 65, no. 3, pp. 940–951, Mar. 2016.

[22] N. Kim, J. Ahn, W. Seo, and K. Choi, "Energy-efficient exclusive last-level hybrid caches consisting of SRAM and STT-RAM," in *Proc. VLSI Syst.-Chip*, Oct. 2015, pp. 183–188.

[23] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proc. Int. Conf. Parallel Archit. Compilation Techn.*, 2008, pp. 72–81.

[24] N. Binkert, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, D. A. Wood, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, and T. Krishna, "The GEM5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.

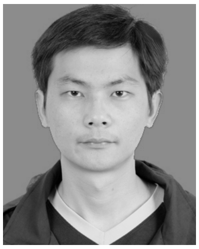
[25] C.-J. Wu, A. Jaleel, W. Hasenplaugh, M. Martonosi, S. C. Steely, Jr., and J. Emer, "Ship: Signature-based hit predictor for high performance caching," in *Proc. Int. Symp. Microarchitecture*, 2011, pp. 430–441.

[26] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0," in *Proc. Int. Symp. Microarchitecture*, 2007, pp. 3–14.

[27] X. Dong, C. Xu, N. Jouppi, and Y. Xie, "NVSim: A circuit-level performance, energy, and area model for emerging non-volatile memory," *Emerging Memory technol.*, pp. 15–50, 2014.

[28] N. Kim, J. Ahn, K. Choi, D. Sanchez, D. Yoo, and S. Ryu, "Benzene: An energy-efficient distributed hybrid cache architecture for manycore systems," *ACM Trans. Archit. Code Optim.*, vol. 15, no. 1, pp. 1–23, Mar. 2018.

[29] J.-H. Choi and G.-H. Park, "NVM way allocation scheme to reduce NVM writes for hybrid cache architecture in chip-multiprocessors," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 10, pp. 2896–2910, Oct. 2017.



FANFAN SHEN received the Ph.D. degree from Wuhan University, China, in 2017. He is currently a full time Lecturer with Nanjing Audit University, China. His main research interests include computer architecture, emerging non-volatile memory, and embedded systems.

CHAO XU received the B.S., M.S., and Ph.D. degrees from the Computer School, Wuhan University, China. He is currently a Professor with Nanjing Audit University, China. His research interests include trusted software and embedded systems.

JUN ZHANG received the Ph.D. degree from the Computer School, Wuhan University, China. He is currently an Associate Professor with the East China University of Technology, Nanchang, China. His main research interests include computer architecture, high performance computing, and embedded systems.