International Journal of Smart Home
Vol. 13, No. 2 (2019), pp.11-16
http://dx.doi.org/10.21742/ijsh.2019.13.2.02

Low-End Memory Subsystem Optimization Process

Jungseok Cho[1], Jeongdu Lee[2],Doosan Cho[3]

EE, Sunchon National University, South Korea
[3]dscho@scnu.ac.kr

Abstract

Traditional memories can be categorized to its characteristics like shortcomings advantages. DRAM and SRAM are volatile memories. SRAMs are faster than DRAMs, but they are used in cache memories because they have low integration and high energy consumption compared to DRAMs. On the other hand, DRAM has a slower processing speed than SRAM, but has high integration and low energy consumption. Since the flash memory is a nonvolatile memory, it is used in portable devices. However, it is slower than DRAM and consumes much energy. Non-volatile memory is a new memory that is attracting attention as the next generation memory. Non-volatile memory is a memory that has both higher integration density than DRAM, and low energy consumption, high-speed operation of SRAM, and non-volatile nature of flash memory, which is advantage of conventional memory. For this reason, it is expected to replace DRAM, flash memory, and hard disk in the future. However, Non-volatile memory is not enough to replace DRAM. In case of using non-volatile memory, higher energy consumption may consume than using DRAM only. To solve this problem, it is necessary to study hybrid memories using DRAM and non-volatile memory together. Hybrid memory can be a solution to the slow processing speed and high energy consumption of the negative effect from non-volatile memory technique.

Keywords: *Energy consumption, Hybrid memory, Embedded system, Nonvolatile memory, Low-end memory subsystemr.*

1. Introduction

NVM (Non-Volatile Memory) has a disadvantage in that it consumes higher energy than DRAM in write operation. To overcome these shortcomings, many researches on hybrid memory using NVM and DRAM are going on. However, there is a problem in using such a hybrid memory. To use hybrid memory, the compiler must be able to allocate memory to allocate memory separately. However, current compilers do not have a way to distinguish memory. To solve these problems, this paper proposes a memory allocation scheme that can allocate memory. In addition, in hybrid memory use, NVM consumes high energy consumption for DRAM due to the write operation. Therefore, it is necessary to allocate a variable with a large write operation to the DRAM and allocate a variable having a large read operation to the NVM. In this paper, we propose an efficient memory mapping technique, after profiling the number of reading / writing of variables. In this paper, we propose a data assignment technique to minimize energy consumption to support hybrid memory consisting of DRAM and NVM. To solve this problem, virtual memory and physical memory are divided into DRAM and NVM

Article history:
Received (October 9, 2018), Review Result (December 4, 2018), Accepted (June 12, 2019)

areas, and physical memory is determined according to the virtual address value, so that it can be allocated to a desired memory. In order to minimize energy consumption, NVM should be allocated to NVM, and variable with more write should be allocated to DRAM to minimize energy consumption because NVM consumes high energy consumption when using hybrid memory.

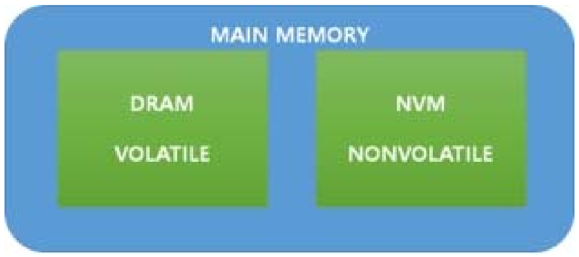

Figure 1. Memory Model

2. Related works

Leakage current is being an important problem in deepening manufacture process, since it is exponentially increased. Nonvolatile memory has characteristics such as nearly zero leakage current and ultralow power consumption. Thus, it can be used as the next generation memory component to solve the scaling problem. Some studies focus on utilization of nonvolatile memory components in multimedia applications.

Zhou et al. [1] proposed an architecture-level technique that can extend lifetime of phase-change memories to an average of 13 to 22 years. Lee et al. [2] mitigated the drawbacks of PCM by analyzing the effects of buffer sizing, row caching, write reduction, and wear leveling (lifetime reduction). They concluded that PCM is a viable candidate to replace DRAM for scalable main memory. Hu et al. [3][4] proposed minimizing write activity in NVM through data migration, scheduling, and recomputation. Shi et al.[5] adopted a smart victim cache to reduce write activity in flash main memory. Catherine H. Gebotys [6] aimed to reduce the number of memory accesses through register allocation in order to reduce the total energy of processor using low power memory.

There are many studies on scratch pad memory as a software-controlled cache. They shows that 60% reduction of power consumption by decreasing main memory accesses[7]. However, the problem of increasing leakage current still remains in the studies. To overcome the problem, non-volatile memory is actively studied, and it is the closest to the next generation memory technology. It consumes nearly zero leakage power, small standby power, and relatively less operation power. There is only concern of the overhead of write operations. In order to solve the write operation overhead of NVM, a hybrid cache is proposed to efficiently utilize the advantages of NVM [8]. Traditional studies focus on reduction of write operations to NVM. Thus, they optimize data assignment to place write intensive data on SPM and read intensive data to NVM.

NVM is attracting attention as the next generation memory because of its lower energy consumption and faster performance than DRAM. However, it has drawbacks of high energy consumption and slow speed in write operation. Many attempts have been made to overcome these drawbacks, and there are many researches on hybrid memory using DRAM and NVM as a solution.

Using the above mentioned NVM to use it with existing memory is called hybrid memory. In this paper, we propose to use hybrid memory as the main memory of personal computer, and the configuration of hybrid memory is applied to STT-MRAM which is one of the most active DRAM and NVM in personal computer. Figure 1 shows the structure of the proposed hybrid memory. Such a hybrid memory type main memory can be a way to minimize energy consumption by taking advantages of DRAM and STTMRAM.

First, there is a study on hybrid memory using DRAM and NVM as main memory. [4] introduced a method of reducing energy consumption of memory by using DRAM and PRAM together. [5] introduced a method of reducing energy consumption by using DRAM and Scratch Pad Memory (SPM) together. [6] introduced a method to save energy by substituting flash memory only for the main memory of a specific application. In addition to main memory, [7] introduced a method to improve performance by using NVM as a file system metadata write buffer.

3. The proposed technique

Figure 1 shows the target memory structure. It uses the conventional traditional memory architecture, except that the main memory area is divided into DRAM and NVM. The cache memory operates in a conventional manner. Since the main memory space is divided into NVM and DRAM, when many read operations are performed on arbitrary variables, they are allocated to the NVM. DRAM has a relatively large number of write operations. The location of these data variables is assigned to the NVM when there is a gain based on the read and write distributions. This is determined by the following formula.

DECISION_FUNCTION 1. ……… (1) EQUATION

NVM_ASSIGNMENT (Energy_READ(data_block(i)) + Energy_WRITE(data_block(i))) >

DRAM_ASSIGNMENT (Energy_READ(data_block(i)) + Energy_WRITE(data_block(i)))

The allocation decision is determined by the compiler and can be applied based on Equation (1). The positioning of the actual data variables must be determined taking into account the actual physical size of each memory. An algorithm that considers the gain and size of data positioning is configured as follows.

```
GREEDY ALGORITHM 1.
INPUT : ALL DATA IN A GIVEN PROGRAM
OUTPUT : DATA PLACEMENT SOLUTION FOR A NVM AND DRAM HYBRID MEMORY
DATA_PLACEMENT(PROGRAM)
{
 FOR_ALL( i = PROGRAM_DATA )
  IF( NVM_SIZE > ( SIZE_SOLUTION_CANDIDATE( DECISION_FUNCTION(i) ) ) )
    ODERED_SOLUTION=ORDERING_NVM_SOLUTION_CANDIDATE(
    SOLUTION_CANDIDATE( ) );
    DECISION_NVM_ASSIGNMENT( ORDERED_SOLUTION );
    ELSE
      DECISION_DRAM_ASSIGNMENT( SOLUTION_CANDIDATE( ) );
}
GRAPH COLORING ALGORITHM 2.
INPUT1 : INTERFERENCE GRAPH for ALL DATA IN A GIVEN PROGRAM
INPUT2 : the NUMBER of COLOR in PARTITIONED NVM SPACE
OUTPUT : DATA PLACEMENT SOLUTION FOR A NVM AND DRAM HYBRID MEMORY
DATA_PLACEMENT(PROGRAM)
{
  WHILE ( !END == VISIT(INTERFERENCE_GRAPH) )
    IF( AVAILABLE_COLOR( ) == TRUE )
      CANDIDATE_NODE = VISIT( INTERFERENCE_GRAPH )
      IF( EDGE_CHECK(CANDIDATE_NODE) )
        ASSIGN_DIFF_COLOR( CANDIDATE_NODE )
      ELSE
        ASSIGN_SAME_COLOR( CANDIDATE_NODE ) }
```

Algorithm 1 shows a greedy approach to a data location algorithm for hybrid memory with DRAM and NVM. It uses all the data used in one program as an algorithm input. The output shows the allocation result of each data variable. Because NVMs have less energy consumption per access than DRAMs, you should make data allocations so that you can read and access data from NVM as much as possible. In the greedy approach, when the variables are placed in the NVM, the gain data is allocated to the NVMs in ascending order of the gain. The remaining data is placed in the DRAM. All data blocks used in one program are the inputs of the algorithm. In this study, each variable is defined as a data block. In Decision_Function shown in Equation (1), it is calculated whether or not there is a gain when it is allocated to NVM for each data i, and if there is a gain, it becomes an NVM allocation candidate. An important part to consider when determining the allocation of data is to check the NVM size limit. Therefore, in order to maximize the gain using the NVM, the gain for each data is stored, and the NVM is allocated after it is sorted in a large order. The ORDER_SOLUTION stores the allocation candidates in order of greatest gain. When the greedy algorithm is allocated in this order and the NVM space is full, the rest of the data is allocated to DRAM in DECISION_DRAM_ASSIGNMENT. Considering the size of the NVM, if it is a gain, it is registered as a solution candidate, and the candidates are sorted according to the gain size and allocated as much as the gain remains in the order of the space. All unallocated data is allocated to DRAM.

We present one more technique for hybrid memory that we will introduce in this study, based on graph coloring. Graph coloring consists of dividing a memory into variable sizes and assigning appropriate variables to each divided space. Resulting in more sophisticated data allocation. We divide the memory by the size of the most commonly used variables, and apply the graph coloring by constructing the interference graph for the variables used in the program. As a result, the limited memory space can be used more efficiently.

4. Experiment

In the experimental section, we will evaluate two techniques for DRAM and NVM to be used in the proposed hybrid memory. We experimented with three multimedia programs, full_search, frame_estimate, field_me. We implemented the in - house memory simulator and compared the proposed two techniques with the DRAM only system. Since the energy consumption of the READ operation is large at NVM, the more effectively the NVM is used, the more energy saving it is. The energy parameters of DRAM and NVM by READ / WRITE operations are 4.4/2.5 and 5.5/12.3. In Figure 3, the baseline is a memory system using DRAM only. Figure 2 shows how the energy consumption of GREEDY ALGORITHM and COLORING algorithm is reduced compared to BASELINE. Based on DRAM only, GREEDY ALGORITHM improved energy consumption by 11.7% on average and COLORING algorithm improved by 32.9% on average. The coloring algorithm was improved by 23.8% compared to the Greedy algorithm. This is because the greedy algorithm is a static technique and the coloring algorithm is a dynamic technique. Because dynamic techniques can place multiple pieces of data in the same space over time, it is possible to allocate more reads to NVM and reduce writes. As a result, 23% energy savings can be achieved.

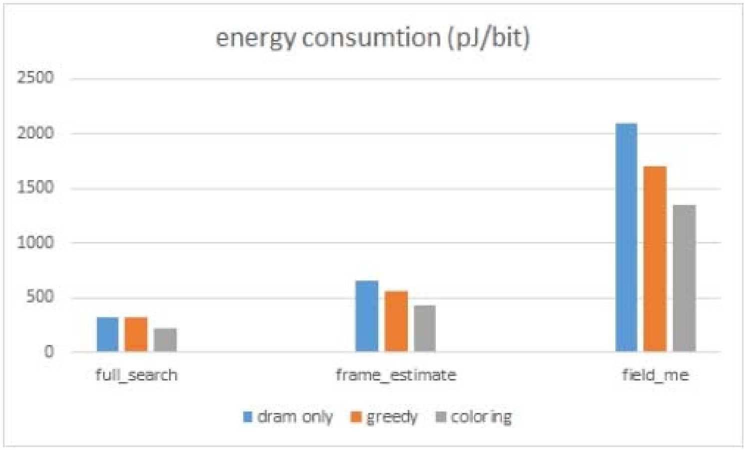

Figure 2. Improvement on energy saving

5. Conclusion

In this study, we proposed two compiler optimization techniques to allocate data to use hybrid memory composed of NVM and DRAM efficiently. The proposed method consists of static and dynamic methods. Applying the proposed techniques to hybrid memory can achieve energy savings of 11% and 32% compared to DRAM only systems. If we apply it to mobile applications that are energy intensive systems, we expect to increase battery life. The proposed technique can be improved in various application environments.

Acknowledgements

This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF - 2018R1D1A1 B07050054).

References

[1] P. Zhou, B. Zhao, J. Yang, and Y.-T. Zhang. “A Durable and Energy Efficient Main Memory Using Phase Change Memory Technology”. Computer Architecture News, pp.14-23 **(2009)**

[2] B. Lee, P. Zhou, J. Yang, Y. Zhang, B. Zhao, E. Ipek, O. Mutlu, D. Burger. “Phase-Change Technology and the Future of Main Memory”. IEEE Micro, Vol.30, No.1 **(2010)**, pp.143. [DOI: 10.1109/MM.2010.24]

[3] J. Hu, C. Xue,W. Tseng, Q. Zhuge and E. H.-M. Sha. “Minimizing Write Activities to Non-volatile Memory via Scheduling and Recomputation”. Proc. 8th IEEE Symposium on Application Specific Processors, **(2010),** pp.7-12. [DOI: 10.1109/SASP.2010.5521139]

[4] J. Hu, C. Xue, W. Tseng, Y. He, M. Qiu and E. H.-M. Sha. “Reducing Write Activities on Non-volatile Memories in Embedded CMPs via Data Migration and Recomputation”. Proceedings 47th IEEE/ACM Design Automation Conference, **(2010)**, pp.350-355. [DOI: 10.1145/1837274.1837363]

[5] L. Shi, C. Xue, J. Hu, W. Tseng and E. H.-M. Sha. “Write Activity Reduction on Flash Main Memory via Smart Victim Cache”. Proceedings ACM/IEEE 20th Great Lakes Symposium on VLSI, **(2010)**, pp. 91-94. [DOI: 10.1145/1785481.1785503]

[6] C. H. Gebotys. “Low Energy Memory and Register Allocation Using Network Flow”. Proceedings of the 34th Annual Design Automation Conference, **(1997)**, pp.435-440

[7] Hyungmin Cho , Bernhard Egger , Jaejin Lee , Heonshik Shin. “Dynamic data scratchpad memory management for a memory subsystem with an MMU”. ACM LCTES, **(2007)**, pp. 13-15. [DOI: 10.1145/1273444.1254804]

[8] S. Kang and A. G. Dean. “Leveraging both data cache and scratchpad memory through synergetic data allocation”. IEEE Real-Time and Embedded Technology and Applications Symposium, (2012) [DOI: 10.1109/RTAS.2012.22]

Authors

Jungseok Cho

Jungseok Cho received his Ph.D. degrees in EE from Sunchon National University(SCNU), Korea in 2018. His research interests are mobile computing, high performance computing and memory system, with an emphasis on energy-aware and fault-resilient design and optimization.

Jeongdu Lee

Jeongdu Lee received his M.S. degrees in EE from Sunchon National University(SCNU), Korea in 2019. His research interests are Multi-copter flight control and fault diagnosis, redundant flight control computer system design.

Doosan Cho

Prof. Cho received his Ph.D. degrees in EECS from Seoul National University, Korea in 2009. He joined Sunchon National University (SCNU) in 2010. His research interests are broadly in the areas of embedded systems, mobile computing, high performance computing, and Security, with an emphasis on energy-aware and fault-resilient design and optimization.