Hybrid Access in Storage-class Memory-aware Low Power Virtual Memory System

Yusuke Shirota, Satoshi Shirai, Tatsunori Kanai
Corporate Research & Development Center, Toshiba Corporation
Tel: +81-44-549-2236, Fax: +81-44-520-1841, E-mail: yusuke1.shirota@toshiba.co.jp

Abstract: With the rapidly growing demands for large capacity main memory in server systems and embedded systems, current DRAM-only approach is hitting the limit due to DRAM's capacity scaling issue and significant background power. With the emergence of storage-class memories (SCMs), we can explore low power, high speed, cheap, and high capacity unified memory system by redesigning virtual memory system in order to efficiently manage the new memory hierarchy of SCM/DRAM. In this paper, we propose *hybrid access* which is a memory hierarchical control method that adaptively switches between SCM-aware low power *aggressive paging* (AP) with small DRAM as cache and *direct access* (DA) to memory bus attached byte-addressable SCM according to data access patterns. Furthermore, we propose an auto-tuning framework that dynamically predicts the optimal control and the optimal DRAM size when AP is selected, based only on time series performance data that can be collected at low cost, using optimal control prediction model generated by machine learning (ML). We show that *hybrid access* has potential to realize efficient unified main memory applicable to a wide range of data access patterns with modest size DRAM.

(Keywords: storage-class memory (SCM), low power, operating system, machine learning)

Introduction: Although in-memory processing can enhance performance for emerging applications such as CPS/IoT, Big Data analytics, and deep learning(AI), increasing DRAM size comes with huge background power. In the meantime, high capacity NVM technologies such as MRAM, PCM, ReRAM are emerging. Expanding main memory requires dynamically optimizing complex data placement so that a fast DRAM can be efficiently integrated in addition to a slower SCM so that performance can be boosted in power efficient manner. Furthermore, for ease of programming, auto-tuning of memory management is important.

In previous works, virtual memory systems with SCM-based swap systems [1, 2] have been proposed to integrate SCM and DRAM to expand main memory size and to provide single image view for ease of programming. AP in low power virtual memory system proposed in [2], aggressively swaps out pages from DRAM to SCM, minimizes DRAM size to the extent of acceptable performance degradation attributed to swapping overhead, and reduces DRAM background power by powering off unused space. Although significant power reduction of memory subsystem can be achieved by efficiently managing unified main memory with modest DRAM size using AP, the performance is limited for applications with extremely low-locality data access patterns.

To address such problems, we introduce *hybrid access*, which selectively uses SCM-aware low power AP method [2] and DA method according to the data access pattern, for efficient unified memory management in low power virtual memory system.

Hybrid Access for SCM-aware Virtual Memory System: In SCM/DRAM main memory, efficiency is improved by distributing the location of data according to their characteristics. It is desirable that data with high locality be accessed on DRAM after the data is copied from SCM to DRAM at page granularity. On the other hand, data with low locality should be kept on memory bus attached SCM and accessed directly. Thus *hybrid access* adopts different access methods, AP and DA, for each data type.

Deficiency of AP can be complemented by leveraging the byte-addressability that SCM offers. In DA, processor can directly access SCM via load/store instructions at cache line granularity. Although memory access latency is relatively high, it can improve overall performance as it can only retrieve the necessary minimum cache lines compared to AP, which suffers significant performance degradation due to DRAM contamination and the resulting frequent page swapping.

ML-based Auto-tuning Framework for Memory Control: We propose an auto-tuning framework, depicted in Fig. 1, for SCM/DRAM main memory control optimization using ML over system level time series performance data. The framework is applied to *hybrid access* control, and the two important tuning parameters, that is (1) the switching timing between AP and DA and (2) the DRAM size to use when AP is applied, are predicted periodically during execution.

***Offline training phase*:** The working set size (WSS), which is a key memory access characteristic that correlates with the two parameters, dynamically changes depending on the application behavior, thus we need to precisely derive them to utilize for performance tuning. However, since deriving WSS requires expensive page reference flag tracking, it is extremely difficult to do it periodically in real time for Big Data applications. Therefore, supervised learning is used to explore the relationship between the WSS and time-series performance data, which includes performance counter data and OS statistical information, that can be acquired at low cost during execution. Then an optimal control prediction model is generated.

 978-1-7281-1749-2/19/$31.00

***Online inference phase*:** The WSS is predicted by applying performance data collected during execution of the target application to the prediction model, and then the two parameters are determined as follows. As shown in Fig. 2, the optimal method is determined by comparing the ratio of the predicted WSS to the resident set size (RSS) with a threshold. For example, when the ratio is smaller, it indicates that memory access is concentrated in a small part of all the memory region that may be used in the application (that is, RSS), and thus AP is selected to exploit locality. Next, when AP is predicted to be the optimal method, the model uses the predicted WSS to adjust the DRAM size available to the application. The remaining DRAM space is set to a low power consumption mode such as power off or self-refresh to reduce power.

Evaluation: We evaluated the proposed method on 2.1GHz *Intel® Xeon® processor E7-8870 v3 with DDR4-2400 memory, running Linux 4.4.0-134-generic. Two benchmarks included in PARSEC [3] with distinct memory access characteristics are used to explain evaluation details: facesim (FS) with high locality and canneal (CN) with extremely low locality. Since SCMs are not commercially available yet, we developed a new performance emulation platform incorporating SCM latency emulation mechanism [4] to estimate the execution time when applying AP and DA in SCM/DRAM main memory environment from a behavior executed on a DRAM machine. SCMs with two different latencies are assumed – each with 10× and 20× the DRAM access latency, described as fast SCM and slow SCM respectively.

First, we evaluated *hybrid access* by deriving the optimal control of each benchmark using our emulation platform. Each row of the emulation result matrix exemplified in Fig. 3 corresponds to each control method (DA at the bottom, AP with DRAM allocation ratio, i.e., DRAM to SCM ratio, ranged from 10%-30%), and each column corresponds to each segment of the application. The performance based on the optimal control is defined as the performance when the optimal control determined for each segment based on the optimization policy is applied, and is calculated based on this matrix. The policy towards power-efficient memory management is to select smaller DRAM allocation ratio if the difference of execution time is within 5%. Optimal control of FS and CN are depicted in Fig. 4 and Fig. 5, respectively. Optimal controls of FS concentrate on AP with DRAM allocation ratio 30% (AP_30%), while DA is basically selected for CN. Results show that high efficiency can be achieved by adopting an appropriate method with *hybrid access*.

Next, we evaluated the proposed auto-tuning framework. The accuracy of the WSS prediction is evaluated. Fig. 6 depicts the RSS and the measured WSS of the two benchmarks. WSS was derived by performing naive reference flag tracking. Fig. 7 depicts the results of WSS prediction. RSS and eight hardware counters related to TLB miss (i.e., DTLB_LOAD/STORE_MISSES.MISS_CAUSES_A_WALK, (DTLB_LOAD/STORE_MISSES.STLB_HIT, PAGE_WALKER_LOADS.DTLB_L1/L2/L3/MEMORY), which have high correlation with WSS, and which are not affected by the method selected at the time of prediction, were utilized as predictors for linear regression prediction. Comparing the two times series data of WSS in Fig. 6 and Fig. 7, we can see that WSSs are accurately predicted throughout program execution.

We then compared the performance applying control method determined using our model to that of optimal control. Since AP should be selected only when reasonable performance enhancement corresponding to the DRAM size to use is obtainable, threshold is set to 1/3 of RSS. For FS, Fig.8 shows the result of comparing the cumulative values of the DRAM sizes used in each segment. Our model (Predicted WSS) was successful in selecting mainly AP_30% which is the optimal method for many segments as depicted in Fig.4, and thus not selecting AP_10% or DA led to obtaining execution time close to the optimal control (Optimal) as depicted in Fig. 9. Results show that our model can reduce DRAM size significantly compared to executing only with DRAM(ALL DRAM) to the extent of acceptable performance degradation. As for CN, our model was successful in selecting optimal method, that is DA as in Fig. 5. As a result, serious performance degradation due to selecting AP (execution time increases by up to 40× as compared with the case of executing only with DRAM) was suppressed and unnecessary DRAM usage was avoided.

Conclusion: We proposed *hybrid access* for SCM/DRAM main memory and developed a ML-based auto-tuning framework that predicts optimal configuration, and confirmed its effectiveness.

[1] Y. Park and H. Bahn, "Efficient management of PCM-based swap systems with a small page size," Journal of Semiconductor Technology and Science, vol. 15, no. 5, pp. 476-484, 2015.

[2] Y. Shirota, S. Yoshimura, S. Shirai, T. Kanai, "Powering-off DRAM with aggressive page-out to storage-class memory in low power virtual memory system," In Proceedings of IEEE COOL Chips XIX, pp. 1-3, 2016.

[3] PARSEC benchmark: http://parsec.cs.princeton.edu/.

[4] H. Volos, G. Magalhaes, L. Cherkasova and J. Li, "Quartz: A Lightweight Performance Emulator for Persistent Memory Software," In Proceedings of the 16th Annual Middleware Conference, pp. 37-49, 2015.

Acknowledgment: This work was based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

978-1-7281-1749-2/19/$31.00 ©2019 IEEE

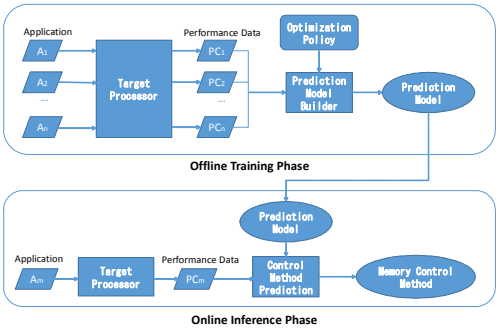


Fig. 1 Memory control optimization framework.

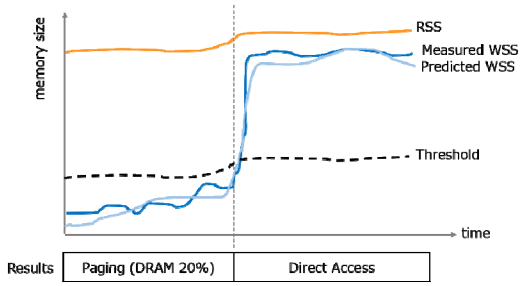


Fig. 2 Selecting optimal control method.

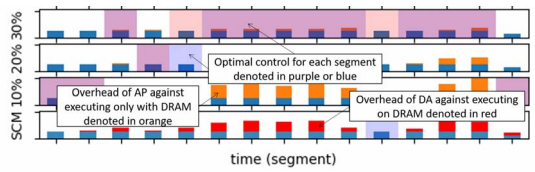


Fig. 3 Optimal control method selection. Vertical bar shows the estimated exec time for each segment.

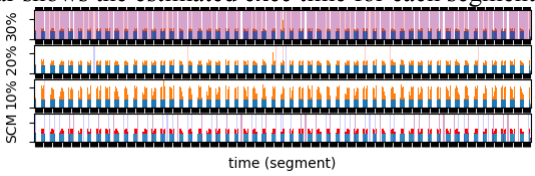


Fig. 4 Optimal control (facesim).

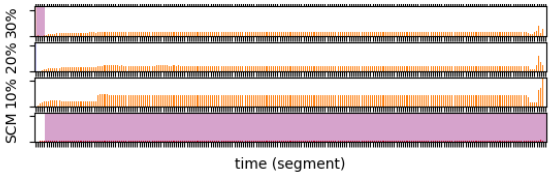
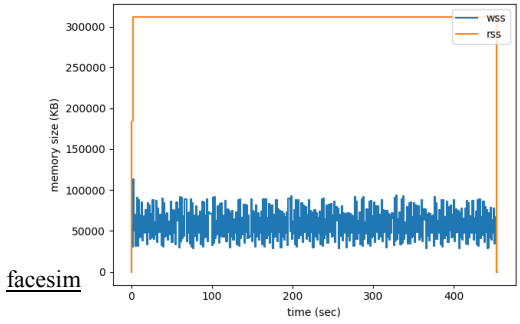


Fig. 5 Optimal control (canneal).



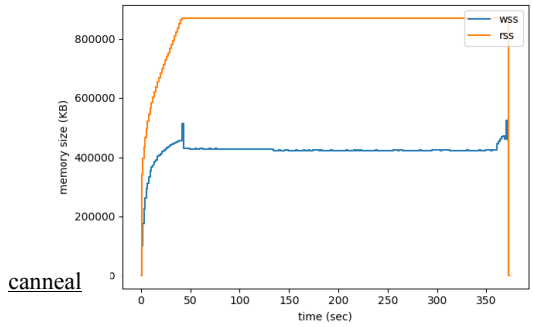
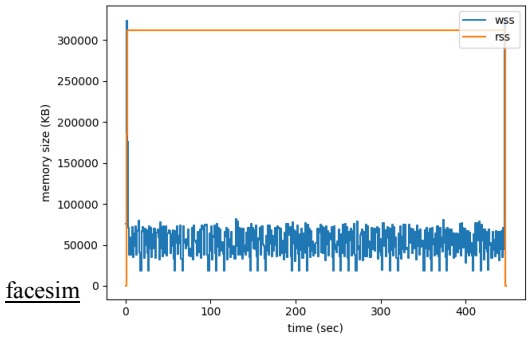


Fig. 6 Measured WSS (facesim, canneal).



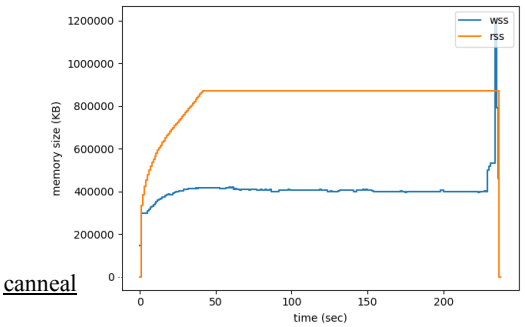


Fig. 7 Predicted WSS (facesim, canneal).

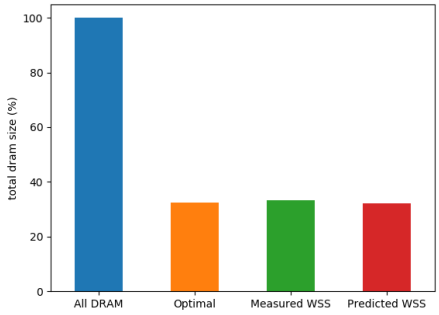


Fig. 8 Comparison of DRAM size (facesim).

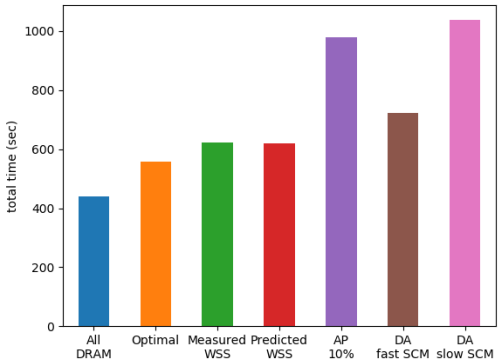


Fig. 9 Comparison of execution time (facesim).

978-1-7281-1749-2/19/$31.00 ©2019 IEEE