Dynamic Hot-Cold Separation Scheme on the Log-structured File System for CE Devices

Dong Hyun Kang† and Young Ik Eom†
†Sungkyunkwan University, South Korea
{kkangsu,yieom}@skku.edu

***Abstract*—Recently, many researchers have focused on the log-structured file system because its layout helps to enhance the performance of the consumer electronics (CE) devices that employ NAND flash storages as their main storage. In this paper, we present a novel hot-cold separation scheme that dynamically identifies the temperatures of blocks in the file system with a hint from the page cache. Our experimental results clearly show that our scheme reduces the number of copy operations by up to 6.1 times compared with the F2FS.**

I. Introduction

Today, the consumer electronics (CE) devices, such as smartphones, smart tablets, and smart TVs, have become an essential part of our life. Most CE devices use NAND flash storages (e.g., eMMCs, SD Cards, and SSDs) as their primary storage media because the flash storages provide attractive features, such as low power consumption, small size, and high performance. In addition, many researchers and engineers enhanced the CE devices in terms of performance and energy efficiency by revisiting traditional software layers, which were designed for desktop and server systems.

Flash-friendly file system (F2FS) was designed based on the log-structured file system and it is one of the enhanced software for the CE devices [1]. In order to take the performance benefits of the flash storage, the file system reshapes random write requests issued to the underlying flash storage into sequential ones with the *append-only approach*. However, this append-only approach requires segment cleaning on the file system to reclaim old blocks, which are no longer to be accessed. Moreover, this cleaning operation is considered harmful since the operation can cause a large number of concurrent read and write requests for copying valid blocks belonging to a victim segment to somewhere else. In order to reduce the number of copy operations, some researchers proposed the hot-cold separation scheme for the log-structured file system [1]–[4]. Unfortunately, the log-structured file system still suffers from the excessive copy operations caused by the segment cleaning because prior work inefficiently classifies blocks of the file system as either hot or cold.

In this paper, we propose a novel hot-cold separation scheme that dynamically identifies the temperatures of blocks (e.g., hot and cold) on the log-structured file system. The proposed scheme employs a hint of the page cache to reduce the number of copy operations during the segment cleaning. We implemented our scheme on the F2FS file system, which is an up-to-date log-structured file system, and compared it with the original F2FS in terms of segment cleaning. Our experimental results show that the proposed scheme efficiently reduces the number of copy operations by up to 6.1 times for real workloads.

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.R0126-15-1082, Management of Developing ICBMS(IoT, Cloud, Bigdata, Mobile, Security) Core Technologies and Development of Exascale Cloud Storage Technology). Young Ik Eom is the corresponding author of this paper.

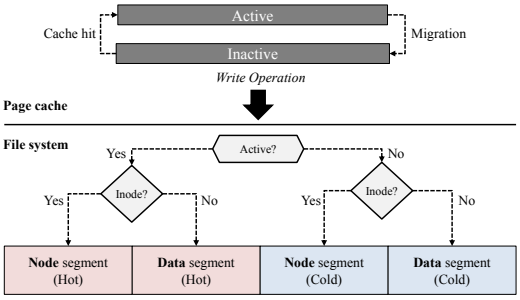


Fig. 1: The overall flow diagram of our scheme

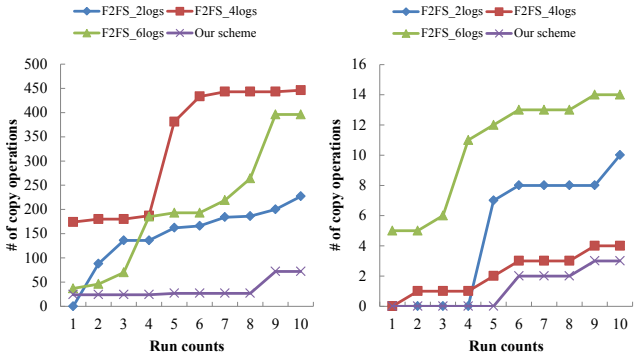
II. Background

Although the log-structured file system was designed for hard disk drives (HDDs), many researchers have focused on the layout of the file system in accordance with the characteristics of NAND flash storages. Generally, the log-structured file system has two key design features. First, the file system divides the whole space of storage into *segments*, and each segment consists of some consecutive 4KB blocks. Second, it employs the *append-only approach*, which sequentially appends a new block rather than overwrites an existing block, to reshape random write requests to sequential ones. Unfortunately, the append-only approach should periodically trigger the segment cleaning operation to reclaim free space. This operation leads to performance drop because it may issue a large number of concurrent read and write requests for copying valid blocks belonging to a victim segment to a new segment. In order to reduce the number of copy operations, many researchers have focused on handling blocks of the file system and determining victim blocks to clean. Some research groups introduced the dynamic hot-cold separation schemes by using hints from the page cache [2]–[4]. However, they do not consider the features of the CE devices, such as low performance and limited memory capacity, which are distinct from desktop and server environments. Recently, for CE devices, some researchers proposed a log-structured file system with the hot and cold separation scheme, F2FS, but their scheme is inefficient because it statically defines blocks as either hot or cold [1].

III. Design and Implementation

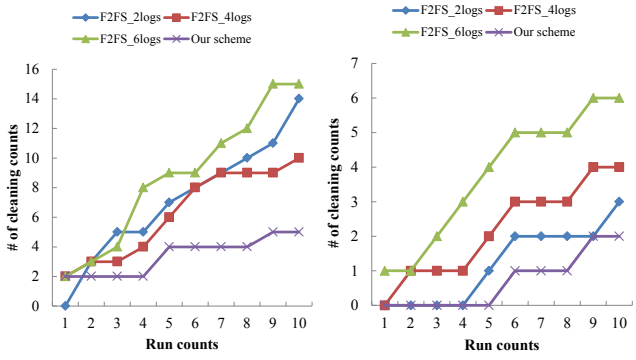
A. Dynamic Hot-Cold Separation Scheme

In this paper, we present a novel hot-cold separation scheme for the log-structured file system. Our scheme is very simple



(a) Fileserver (b) Mobile (Gmail)

Fig. 2: The number of copy operations



(a) Fileserver (b) Mobile (Gmail)

Fig. 3: The number of segment cleaning operations

and straightforward, however it significantly reduces the number of copy operations during the segment cleaning.

Figure 1 shows the overall flow diagram of our scheme. In order to reduce the number of copy operations, we dynamically identify the temperatures of blocks (e.g., hot and cold) with semantic information of the *page cache*. The page cache is commonly used to mitigate performance gap between CPUs and the underlying storage devices, and keeps page references in a temporal order (i.e., temporal locality) to avoid frequent storage accesses caused by page cache miss. Therefore, it already knows which pages are hot or cold based on the position in the page cache: hot pages are kept on an active list, while cold pages are kept on an inactive list. We gracefully employ this positional information to separate blocks of the log-structured file system as hot or cold. As mentioned before, since the log-structured file system basically appends a new block rather than overwrites an existing block, it always requires a block allocation to persistently store contents of an updated page. Therefore, when a block allocation is triggered, we first check the location of the page on the page cache. If the page exists on the active list, we assign a new block in a *hot segment* because the page has been frequently referenced and may potentially be overwritten in the near future. Otherwise, we assign a new block in a *cold segment*. As a result, most blocks in a *hot segment* will be quickly invalidated while blocks that in a *cold segment* will be slowly invalidated.

B. Implementation

For evaluation, we implemented our scheme on F2FS [1], a well-known file system for CE devices. F2FS also classifies storage space into two groups, such as node and data segments, and it statically characterizes each segment according to three temperature levels (e.g., hot, warm, and cold). In order to retrofit our scheme, we modified its temperature level and block allocation mechanism. In our implementation, we employed Ubuntu 12.04 with Linux kernel 4.4.10. All experiments were carried out on a machine configured with a 3.5 GHz Intel Core i7 with 8 GB of physical memory.

IV. Performance Evaluation

In this section, we compare our scheme with the original F2FS [1] in terms of segment cleaning. F2FS was run by varying its log level from two to six since it supports three log levels, such as two, four, and six logs. For extensive evaluation, we used the fileserver workload of Filebench and real-world mobile workload that was collected by running Gmail application on Nexus 6.

Figure 2 and Figure 3 show the results of ten consecutive runs to understand how the segment cleaning is performed. As expected, our scheme significantly outperforms the F2FS in most cases. As shown in Figure 2, our scheme significantly reduces the total number of copy operations compared with the original F2FS: by up to 6.1 times for fileserver and 4.6 times for mobile workload. This results mean that blocks in a segment have the same temperature.

We also measured the number of segment cleaning operations (i.e., cleaning count) to deeply understand the relationship between block temperatures and segment cleaning. In Figure 3, our scheme shows the lowest cleaning counts in most cases and the gap becomes considerably large as the run count increases. In the best case, our scheme eliminates triggering of the segment cleaning by up to 3 times compared with the original F2FS. The major reason of this results is that the number of cleaning operations does not increase when all blocks belonging to a segment are invalidated, simultaneously (i.e., the same temperature).

V. Conclusion

In this paper, we proposed the dynamic hot-cold separation scheme for the log-structured file system, which positively affects the overall performance of CE devices by reducing a large number of concurrent read and write operations caused by segment cleaning. Our experimental results clearly show that our scheme outperforms the state-of-the-art log-structured file system, F2FS. We also believe that the key idea of our scheme can be widely adopted for other environments because the need for log-structured file system is gradually growing.

References

[1] C. Lee, D. Sim, J.-Y. Hwang, and S. Cho, "F2FS: A New File System for Flash Storage," in *Proc. of the 13th USENIX File and Storage Technologies (FAST)*, 2015.

[2] J. Wang and Y. Hu, "WOLF-A Novel Reordering Write Buffer to Boost the Performance of Log-Structured File Systems," in *Proc. of the 1st USENIX File and Storage Technologies (FAST)*, 2002.

[3] W. Wang, Y. Zhao, and R. Bunt, "HyLog: A High Performance Approach to Managing Disk Layout," in *Proc. of the 3rd USENIX File and Storage Technologies (FAST)*, 2004.

[4] C. Min, K. Kim, H. Cho, S.-W. Lee, and Y. I. Eom, "SFS: Random Write Considered Harmful in Solid State Drives," in *Proc. of the 10th USENIX File and Storage Technologies (FAST)*, 2012.