

낸드 플래시 메모리 기반 저장장치의 가비지 컬렉션 최적화를 위한 효율적인 TRIM 스케줄링 기법

김병국¹, 류갑상^{2*}

¹동신대학교 컴퓨터학과 박사과정, ²동신대학교 컴퓨터학과 교수

An Efficient TRIM Scheduling Method for Optimizing Garbage Collection in NAND Flash-Based Storage Devices

Byung-Gook Kim¹, Gab-Sang Ryu^{2*}

¹Doctor's Course, Student, Dept. of Computer Science, Dongshin University

²Professor, Dept. of Computer Science, Dongshin University

요약 본 논문은 NAND 플래시 메모리 기반 저장장치에서 TRIM 명령을 포함한 입출력 명령어의 동적 우선순위 스케줄링 기법을 적용하여 가비지 컬렉션(Garbage Collection)의 성능을 최적화하는 방법에 대해 다룬다. ATA 표준에 추가된 TRIM 명령은 운영체제가 저장장치에 더 이상 사용하지 않는 데이터 블록의 정보를 전달함으로써, 저장장치가 해당 블록을 사전에 정리하고 새로운 데이터를 저장할 수 있는 가용 블록을 확보할 수 있도록 지원한다. 그러나 TRIM 명령의 처리 과정은 읽기 및 쓰기 명령의 성능에 영향을 미치므로 일반적으로 낮은 우선순위로 처리된다. 한편, NAND 플래시 메모리 기반 저장장치가 지속 성능(Sustained Performance) 상태에 진입하면 가용 블록의 부족으로 인해 가비지 컬렉션에 소요되는 비용이 급격히 증가하는 특성이 나타난다. 이에 본 연구에서는 지속 성능 상태에서 TRIM 명령의 우선순위를 동적으로 조정하는 스케줄링 기법을 통해 가비지 컬렉션의 성능을 향상시키는 방안을 제안하고 가비지 컬렉션 성능에 미치는 긍정적인 영향을 확인한다.

주제어 : 낸드 플래시 메모리, 저장시스템, TRIM 명령, 가비지 컬렉션, 동적 우선순위 스케줄링

Abstract This paper presents a method to optimize garbage collection (GC) performance in NAND flash-based storage devices by applying a dynamic priority scheduling technique for input/output commands, including the TRIM command. The TRIM command, introduced in the ATA standard, enables the operating system to inform the storage device of data blocks that are no longer in use. This allows the device to proactively erase those blocks and secure available space for future writes. However, because the processing of TRIM commands may interfere with the performance of read and write operations, it is typically treated with low priority. Meanwhile, when a NAND flash-based storage device enters a sustained performance state, the cost of garbage collection increases sharply due to a shortage of available blocks. To address this issue, this study proposes a dynamic priority scheduling scheme that adjusts the priority of TRIM commands during the sustained performance phase. Experimental results demonstrate that the proposed approach improves garbage collection efficiency and has a positive impact on overall system performance.

Key Words : NAND Flash Memory, Storage System, TRIM Command, Garbage Collection, Dynamic Priority Scheduling

*교신저자 : 류갑상(gryu@dsu.ac.kr)

접수일: 2025년 04월 22일 수정일 2025년 05월 13일 심사완료일 2025년 06월 02일

1. 서론

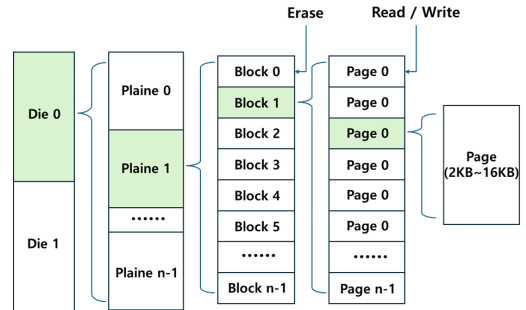
NAND 플래시 메모리는 비휘발성 저장장치로서 SSD (Solid State Drive)의 핵심 구성 요소로 널리 사용되고 있으며, 고속 데이터 접근성과 낮은 전력 소모를 바탕으로 기존의 HDD(Hard Disk Drive)를 빠르게 대체하고 있다[1-3]. 그러나 NAND 플래시의 물리적 특성상 데이터의 덮어쓰기가 불가능하고, 새로운 데이터를 저장하려면 기존 블록을 삭제해야 한다. 이러한 제약사항으로 인해 SSD는 불필요한 데이터를 정리하는 가비지 컬렉션 (Garbage Collection, GC)과정을 필수적으로 수행하게 된다[4-6]. 저장장치의 여유 공간(Over-Provisioned Blocks)이 줄어들수록, FTL(Flash Translation Layer)에서 활용할 수 있는 가용 블록 수가 감소하고, 가비지 컬렉션으로 인한 성능 저하가 더욱 두드러지게 나타난다. 특히, 고부하 워크로드(Heavy Workload) 환경에서는 SSD가 지속 성능(Sustained Performance) 상태에 진입하며, 가비지 컬렉션의 처리 비용이 급격히 상승하는 현상이 발생한다. 이러한 문제를 개선하기 위해 ATA 표준에서는 TRIM 명령을 도입하여 운영체제가 SSD에 게 더 이상 사용하지 않는 데이터 블록의 정보를 전달할 수 있도록 지원한다. 이를 통해 SSD는 불필요한 데이터를 사전에 정리하고 새로운 데이터를 저장할 수 있는 가용 블록을 확보할 수 있다. 그러나 TRIM 명령은 일반적으로 낮은 우선순위로 처리되며, 처리 중에는 읽기 및 쓰기 명령의 성능에 영향을 미칠 수 있다. 본 연구에서는 SSD가 지속 성능 상태에 진입했을 때 발생하는 가비지 컬렉션 성능 저하 문제를 해결하기 위해, TRIM 명령의 우선순위를 동적으로 조정하는 스케줄링 기법을 제안한다. 제안된 기법은 가용 블록의 부족 상황에서 TRIM 명령의 처리 우선도를 높여 가비지 컬렉션 효율을 개선하는 것을 목표로 한다.

2. 배경

2.1 NAND 플래시 메모리의 구조와 특징

Fig. 1은 NAND 플래시 메모리의 구조를 보여주고 있다. NAND 플래시 메모리는 셀(Cell), 페이지(Page), 블록(Block), 플레인(Plane) 그리고 다이(Die) 등의 계층적 구조로 구성된다. 하나의 플레인은 여러 블록으로 구성되어 있고, 블록은 다시 여러 페이지로 구성되어 있다. 블록은 수십에서 수백 개의 페이지로 구성되며, 삭제

(Erase) 연산은 블록 단위로만 가능하다.



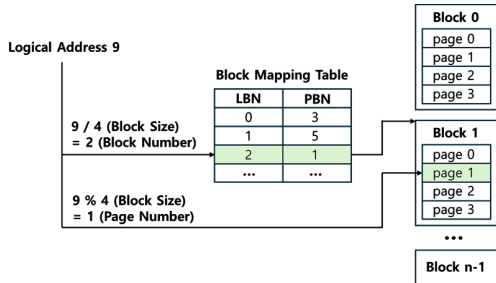
[Fig. 1] Structure of NAND Flash Memory

이로 인해 플래시 메모리는 기존 데이터를 직접 덮어 쓸 수 없고, 새로운 데이터를 쓰기 전 반드시 블록을 삭제해야 하는(Erase-Before-Write) 제약을 갖는다. 페이지는 일반적으로 2KB에서 16KB 크기의 데이터를 저장할 수 있는 단위로 구성되어 있으며, 읽기(Read) 및 쓰기(Write) 연산은 페이지 단위로 수행된다. 이러한 물리적 제약을 극복하고, 일반 저장장치처럼 읽기, 쓰기, 삭제가 가능하도록 추상화하기 위해서 FTL을 사용한다. FTL은 논리 블록 주소(Logical Block Address)와 물리 블록 주소(Physical Block Address) 간의 매핑(Mapping)을 관리하여, 논리 주소를 실제 물리 주소로 변환해 주는 역할을 한다[7-11]. 성능 관점의 특징으로는 읽기 및 TRIM 연산 대비 쓰기 연산의 평균 지연 시간이 약 10배 정도 느리고, 삭제 연산은 약 100배 정도 느리다. 따라서 쓰기 및 삭제 연산이 수반되는 가비지 컬렉션의 효율을 개선하기 위한 많은 연구가 진행되고 있다. TRIM 명령은 가비지 컬렉션 효율을 개선하기 위한 목적으로 수행한다.

2.2 FTL 매핑 기법

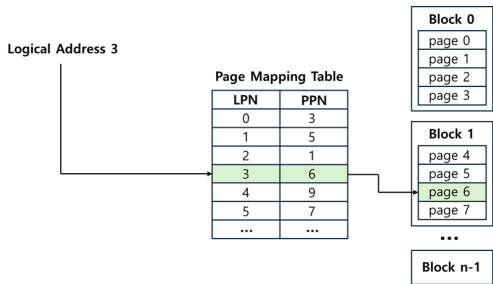
NAND 플래시 메모리는 쓰기 및 삭제 연산에 있어 구조적인 제약이 존재한다. 특히, 데이터를 덮어쓰는 것이 불가능하며, 기존 데이터를 수정하기 위해서는 해당 블록 전체를 소거해야 하는 특성이 있다. 이러한 제약을 극복하기 위하여 FTL이 도입되었다. FTL은 논리 주소와 물리 주소 간의 매핑을 담당하며, 가비지 컬렉션, 웨어 레벨링(Wear Leveling), 불량 블록 관리 등의 기능을 통해 플래시 메모리의 수명 및 성능을 보장한다. FTL의 핵심 기능 중 하나는 주소 매핑(Address Mapping) 기법으로, 이는 논리 주소를 실제 플래시 메모리 상의 물리 주소로 변환하는 방법을 의미한다. 일반적으로 블록 단

위 매핑(Block-level Mapping), 페이지 단위 매핑(Page-level Mapping), 그리고 하이브리드 매핑(Hybrid Mapping)의 세 가지로 분류된다.



[Fig. 2] Block Mapping Algorithm

Fig. 2는 블록 단위 매핑을 위한 알고리즘을 보여주고 있다. 블록 단위 매핑은 논리 블록을 물리 블록에 1:1로 대응시키는 방식으로, 구현이 간단하고 매핑 테이블 크기가 작다는 장점이 있다. 그러나 일부 데이터를 수정할 경우 전체 블록을 새로운 블록으로 복사해야 하므로 쓰기 증폭(Write Amplification)이 크고 성능이 저하되는 문제가 발생한다. 이에 따라, 보다 세분화된 단위로 매핑을 수행하는 페이지 단위 매핑이 제안되었다.



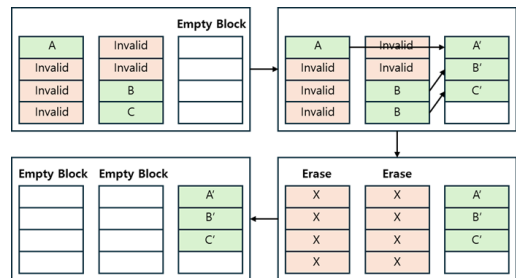
[Fig. 3] Page Mapping Algorithm

Fig. 3는 페이지 단위 매핑을 위한 알고리즘을 보여주고 있다. 페이지 단위 매핑은 논리 페이지를 개별 물리 페이지로 직접 매핑함으로써 유연성과 성능을 크게 향상시킨다. 특히, 업데이트 연산 시 기존 데이터를 무시하고 새로운 페이지에 데이터를 기록할 수 있어 효율적이다. 그러나 이 방식은 매핑 테이블의 크기가 매우 커지므로, 전체 테이블을 메모리에 상주시키기 위해 상당한 양의 DRAM이 필요하다는 단점이 존재한다. 이러한 단점을 보완하기 위하여 블록 단위와 페이지 단위의 장점을 절충한 하이브리드 매핑 기법이 도입되었다. 하이브리드 방식은 일반적으로 데이터를 블록 단위로 저장하되, 일부 갱신된 데이터를 위한 로그 블록을 별도로 운영함으

로써 성능과 공간 효율성을 동시에 확보하고자 한다.

2.3 가비지 컬렉션과 쓰기 증폭

SSD 등 NAND 플래시 메모리를 기반으로 한 저장장치에서는 기존의 자기 디스크 기반 저장장치와는 다른 방식으로 데이터를 저장하고 관리한다. 특히, NAND 플래시는 데이터를 덮어쓰기 할 수 없고, 반드시 블록 단위로 삭제 연산을 수행해야 쓰기 연산이 가능하다. 이러한 제약 극복을 위해 FTL을 도입하고 논리 주소와 물리 주소 간의 매핑을 수행한다. FTL의 기능 중 하나인 가비지 컬렉션은 유효하지 않은 페이지를 포함하는 블록을 식별하고, 그 블록에 남아 있는 유효 페이지를 다른 블록으로 이동시킨 뒤, 해당 블록을 지우는 과정을 수행한다. 이를 통해 SSD는 새로운 데이터를 기록할 수 있는 여유 공간을 확보한다. 가비지 컬렉션은 저장장치의 성능과 수명에 밀접한 영향을 미친다. 유효 데이터를 복사하고 블록을 지우는 작업은 쓰기 증폭을 유발할 수 있으며, 이는 저장장치의 성능 저하와 수명 단축의 원인이 된다 [12-13]. 따라서 효율적인 가비지 컬렉션 알고리즘을 설계하는 것은 저장장치의 성능 최적화에 있어 매우 중요한 연구 과제 중 하나이다.



[Fig. 4] Example of Garbage Collection

Fig. 4는 FTL에서 가용 블록을 확보하기 위한 가비지 컬렉션을 수행하는 예를 보여주고 있다. 가비지 컬렉션에서 가장 중요한 정책 중 하나는 정리 대상 블록을 선택하는 것이다. 대표적인 예로 Greedy 정책이 있으며, 이는 전체 블록 중 유효 페이지 수가 가장 적은 블록을 선택하여 지우는 방식이다. 쓰기 증폭이란 저장장치가 실제 사용자로부터 받은 쓰기 요청보다 내부적으로 더 많은 양의 데이터를 쓰게 되는 현상을 의미한다. 주로 가비지 컬렉션 과정에서 유효 데이터 복사와 블록 삭제 연산으로 발생하며, 저장장치의 성능 저하와 플래시 셀의 수명 단축을 초래한다. 따라서 가비지 컬렉션의 효율을 높이고, 쓰기 증폭을 최소화하는 것은 저장장치 설계에서

매우 중요한 고려 사항이다. 쓰기 증폭을 줄이기 위한 전략에는 오버프로비저닝(Over Provisioning) 영역의 확보 등이 포함된다[14-16]. 또한 TRIM 명령은 SSD가 가비지 컬렉션 및 웨어 레벨링 작업을 더 효율적으로 수행할 수 있도록 하여 쓰기 증폭을 개선할 수 있다. TRIM 명령이 수행되지 않을 경우, SSD는 삭제된 데이터 영역을 여전히 유효한 데이터로 인식할 수 있으며, 이는 불필요한 데이터 유지 및 성능 저하로 이어질 수 있다. 결과적으로 TRIM 명령은 SSD의 내부 처리 효율성을 제고함으로써 쓰기 지연을 감소시키고, 불필요한 데이터 이동을 억제하여 전체적인 시스템 응답 속도와 저장장치의 수명을 향상시키는데 기여한다.

2.4 고부하 워크로드 환경에서의 스케줄링 문제

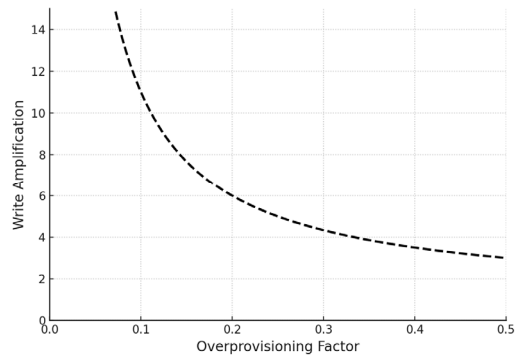
SSD는 높은 입출력 성능을 제공하기 위해 다양한 병렬 처리 메커니즘을 채택하고 있으며, 이 중 명령어 큐잉(Command Queuing)은 I/O 성능 최적화를 위한 핵심 기술이다. 명령어 큐잉은 호스트 시스템이 SSD에 복수의 명령어를 동시에 전달할 수 있도록 허용하고, SSD 컨트롤러가 이를 효율적으로 재정렬하여 내부 자원 활용을 극대화하는 기술이다. SATA 인터페이스는 NCQ(Native Command Queuing)을 통해 명령어 큐잉을 지원하며, 최대 32개의 명령어를 하나의 큐에 저장할 수 있다. NVMe 인터페이스는 SQ(Submission Queue)와 CQ(Completion Queue)로 구성되며, 각각 호스트 메모리에 존재한다. SSD는 DMA(Direct Memory Access)를 통해 해당 큐에 직접 접근할 수 있으며, 최대 64K 개의 큐를 지원하고, 각 큐는 최대 64K 개의 명령어를 수용할 수 있다. SSD 내부의 컨트롤러는 쓰기, 읽기, 가비지 컬렉션, TRIM 등 각 명령어의 우선순위와 NAND 리소스의 상태를 고려하여 최적의 명령 처리 순서를 결정한다. 이때 대부분의 SSD 펌웨어는 TRIM을 배경 작업으로 처리하거나 쓰기 명령보다 낮은 우선순위로 두는 정책을 사용한다. 만약 쓰기 명령을 포함하는 고부하 워크로드가 지속되어 유희 시간이 발생하지 않는다면, 큐잉된 TRIM 명령은 실행이 지연되는 문제가 발생한다.

3. 동적 우선순위 스케줄링 기법을 적용한

TRIM 정책

3.1 TRIM 우선순위 정책

Fig. 5는 오버프로비저닝 인수에 따른 쓰기 증폭 예측 값을 보여주고 있다. 블록 당 페이지 수는 256개로 가정하고, Greedy GC 정책을 기반으로 예측하였다. Greedy GC 정책은 무효화 된 페이지가 가장 많은 블록을 희생 블록으로 선택한다. 따라서 SSD가 지속 성능 상태가 되면 각 블록은 일정한 유효 페이지를 가진다. 이 유효 페이지의 분포를 균등 분포로 근사하고 웨어 레벨링 가중치를 추가하여 유효 페이지 수와 쓰기 증폭을 예측하였다.



[Fig. 5] Write amplification of Greedy GC model

이를 통해 OP 인수가 낮아질수록 쓰기 증폭이 급격하게 증가하는 것을 확인할 수 있다. 이는 저장장치가 지속 성능 상태(Sustained Performance)일 경우 대부분의 블록이 이미 유효 데이터로 채워져 있으며, OP 영역이 작을수록 가비지 컬렉션 수행 시 임시로 쓸 수 있는 대상 블록이 적어 효율이 낮아지기 때문이다. 즉, 많은 유효 데이터를 복사하게 되어, 결과적으로 쓰기 증폭이 급격히 증가하는 경향을 보인다.

<Table 1> SSD Models with Over-Provisioning and Intended Use

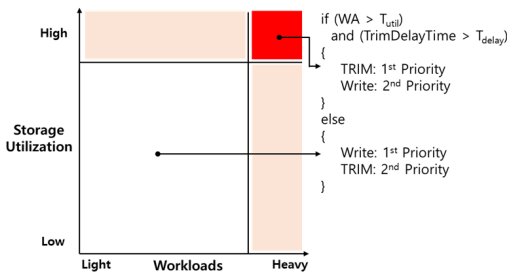
Model	Capacity	OP	Intended Use
860EVO	500GB	6.7%	Client
845DC EVO	400GB	28%	Enterprise
DC S3710	800GB	30~40%	Enterprise
DCS3610	800GB	10~20%	Enterprise
SN740	1TB	7.4%	Client
MX500	500GB	10%	Client
600 Pro	200GB	22%	Enterprise

Table 1에 OP 인수를 포함한 주요 제조사의 SSD 제품 정보를 나열하였다. 클라이언트용 SSD 제품은 대부분

OP 인수가 0.1 이하로 설정되어 있음을 확인할 수 있다. 따라서 지속 성능 상태에서 고부하 워크로드가 지속되면 쓰기 증폭이 급격하게 상승하게 된다. 따라서 쓰기와 TRIM 명령이 혼재하는 고부하 워크로드가 지속되는 상황에서는 쓰기 명령 대비 TRIM 명령의 우선순위를 높여서 가비지 컬렉션 효율을 보다 효과적으로 개선시킬 필요가 있다. 이를 위해서는 명령 스케줄링을 위한 우선순위 정책이 동적으로 적용되어야 한다.

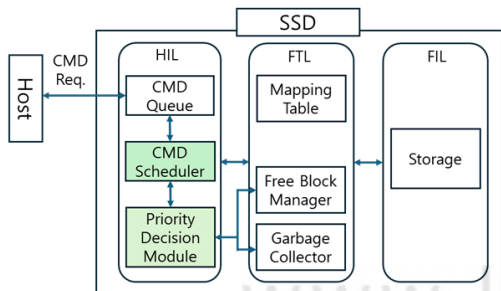
3.2 동적 우선순위 알고리즘을 적용한 스케줄링

Fig. 6은 저장 공간 사용 비율과 워크로드 수준을 기반으로 TRIM 명령에 대한 우선순위의 변경이 필요한 조건을 보여준다.



[Fig. 6] Scope of Dynamic Priority Scheduling

저장 공간 사용 비율이 높지 않다면 고부하 워크로드가 발생하더라도 쓰기 증폭에 대한 영향도가 낮다. 반면에 저장 공간 사용 비율이 높은 상태에서 고부하 워크로드가 발생한다면 OP 인수가 낮을수록 쓰기 증폭이 급격하게 증가하게 된다. 따라서 TRIM 명령의 실행을 위한 우선순위를 쓰기 명령 대비 높여 TRIM 명령이 우선하여 수행되도록 스케줄링한다면 전체적인 성능 향상에 도움이 될 수 있다. 낮은 워크로드에서는 유휴 시간이 발생하게 되어 TRIM 명령이 배경 작업으로 처리될 수 있으므로 우선순위를 기본 정책으로 복원한다. Fig. 7은 SSD에 적용된 동적 우선순위 스케줄링의 구조를 보여준다.

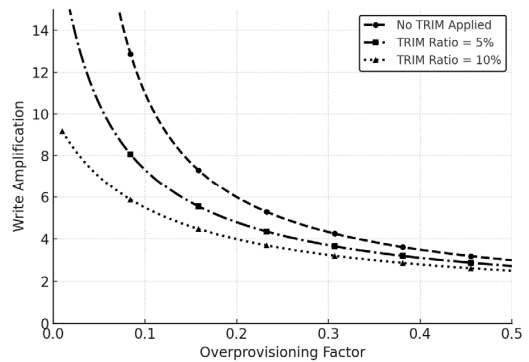


[Fig. 7] Architecture of Dynamic Priority Scheduling

우선순위 결정 모듈(Priority Decision Module)은 쓰기 증폭 값, TRIM 명령의 대기 시간, 자유 블록의 비율 또는 유휴 페이지의 비율 등 관련 요소를 참조하여 하드웨어의 성능 및 펌웨어의 동작 구조에 따라 우선순위의 결정을 위한 최적화된 임계치를 적용할 필요가 있다. 본 연구에서 사용한 쓰기 증폭 예측에 사용된 시뮬레이션 방법은 수식 (1)을 적용하였다.

$$WA = \frac{1}{1 - u_{eff}}, \text{ where } u_{eff} = \frac{1}{1 + OP} \cdot (1 - TRIMRatio) \quad (1)$$

u_{eff} 는 쓰기 증폭 예측에 필요한 유휴 데이터 비율을 의미한다. 가장 효과적인 개선 결과를 확인하기 위하여 지속적인 랜덤 쓰기 명령과 일정한 비율의 순차 TRIM 명령을 포함하는 고부하 워크로드가 지속되어 유휴 시간이 발생하지 않고, TRIM 명령의 우선순위가 쓰기 명령보다 높은 조건임을 가정하였다. 순차 TRIM 명령에 의한 무효 페이지에 의해 발생하는 가용 블록은 GC 알고리즘의 희생 블록(Victim Block)으로 선택될 수 있으므로, u_{eff} 에 TRIM 비율의 영향을 반영하였다. Fig. 8은 위 조건으로 워크로드 상에서 TRIM 비율에 따른 쓰기 증폭의 변화를 시뮬레이션한 결과를 보여준다.



[Fig. 8] WA Variation by TRIM Ratio

일반적인 클라이언트 SSD에 적용되는 OP 인수는 0.1 전후이다. 워크로드에서 랜덤 쓰기 명령만 단독으로 지속되는 경우 OP 인수가 0.1인 경우 쓰기 증폭 값은 11로 시뮬레이션 되었으며, 순차 TRIM 명령의 비율이 5% 포함된 경우, 쓰기 증폭 값은 7.3으로 계산되어 33.6% 개선되었다. 또한 순차 TRIM 명령의 비율이 10% 포함된 경우, 쓰기 증폭 값은 5.5로 계산되어 50% 개선되었다.

4. 결론

본 연구에서는 지속 성능 상태에서 TRIM 명령의 우선순위를 동적으로 조정하는 스케줄링 기법을 통해 가비지 컬렉션 성능을 향상시키는 방안을 제안하고 가비지 컬렉션 성능에 미치는 긍정적인 영향을 확인하였다. 향후 스케줄링 조건에 효율적인 쿼터 개념 등을 추가함으로써, 시스템 전체의 QoS(Quality of Service)를 개선하도록 하고, 다양한 워크로드에 대한 시뮬레이션을 통하여 개선된 정책을 연구할 예정이다.

REFERENCES

- [1] H.S.Lee, "A method for optimizing lifetime prediction of a storage device using the frequency of occurrence of defects in NAND flash memory," *Journal of Internet of Things and Convergence*, Vol.7, No.4, pp.9-14, 2021.
- [2] H.S.Lee, "High Efficiency Life Prediction and Exception Processing Method of NAND Flash Memory-based Storage using Gradient Descent Method," *Journal of Convergence for Information Technology*, Vol.11, No.11, pp.44-50, 2021.
- [3] H.S.Lee, "A Study on Characteristics and Techniques that Affect Data Integrity for Digital Forensic on Flash Memory-Based Storage Devices," *Journal of Internet of Things and Convergence*, Vol.9, No.3, pp.7-12, 2023.
- [4] F.Yu and H.Yan, "An Efficient Hot-Cold Data Separation Garbage Collection Algorithm Based on Logical Interval in NAND Flash-based Consumer Electronics," *IEEE Transactions on Consumer Electronics*, Vol.69, No.3, pp.431-440, 2023.
- [5] S.H.Son and S.Y.Ahn, "Optimizing Garbage Collection Overhead of Host-level Flash Translation Layer for Journaling Filesystems," *Journal of the Institute of Internet, Broadcasting and Communication (JIIBC)*, Vol.13, No.2, pp.27-35, 2021.
- [6] S.Nie, J.Niu, C.Yang, P.Zhang, Q.Wang, D.Wang and W.Wu, "Exploiting Data Duplication to Reduce Data Migration in Garbage Collection for SSDs," *Electronics*, Vol.14, No.9, pp.1873, 2023.
- [7] H.S.Lee, "A Safety IO Throttling Method Inducting Differential End of Life to Improving the Reliability of Big Data Maintenance in the SSD based RAID," *Journal of Digital Convergence*, Vol.20, No.5, pp.593-598, 2022.
- [8] H.S.Lee, "A Study on the Performance Measurement and Analysis on the Virtual Memory based FTL Policy through the Changing Map Data Resource," *Journal of Internet of Things and Convergence*, Vol.9, No.1, pp.71-76, 2023.
- [9] S.S.Chae, R.Mativenga, J.Y.Paik, M.Attique and T.S.Chung, "DSFTL: An Efficient FTL for Flash Memory Based Storage Systems," *Electronics*, Vol.9, No.1, pp.145, 2020.
- [10] W.Xie, Y.Chen and P.C.Poth, "ASA-FTL: An Adaptive Separation Aware Flash Translation Layer for Solid State Drives," *Parallel Computing*, Vol.61, pp.3-17, 2017.
- [11] H.S.Lee, "A Memory Mapping Technique to Reduce Data Retrieval Cost in the Storage Consisting of Multi Memories," *Journal of Internet of Things and Convergence*, Vol.9, No.1, pp.19-24, 2023.
- [12] C.H.Park, Y.J.Won and Y.J.Park, "A Study on Write Amplification Model in Flash Storage Devices," *Journal of The Institute of Electronics and Information Engineers*, Vol.56, No.7, pp.24-30, 2019.
- [13] Y.I.Choi and S.Y.Ahn, "Separating the File System Journal to Reduce Write Amplification of Garbage Collection on ZNS SS," *Journal of Multimedia Information System (JMIS)*, Vol.9, No.4, pp.261-268, 2022.
- [14] P.Patil and B.Jagdale, "Analyzing the Common Host Interfaces Used with SSDs," *Applied GIS*, Vol.4, No.3, pp.1-4, 2016.
- [15] H.Suthar and P.Sharma, "Method for Extracting Data from an Overprovisioned SSD," *2022 IEEE Pune Section International Conference*, pp.1-6, 2022.
- [16] H.S.Lee, "Performance analysis and prediction through various over-provision on NAND flash memory based storage," *Journal of Digital Convergence*, Vol.20, No.3, pp.343-348, 2022.

김 병 국(Byung-Gook Kim)

[중심회원]



- 2006년 7월 ~ 2021년 4월 : 삼성전자 DS부문 책임연구원
- 2022년 8월 ~ 현재 : 배화여자대학교 소프트웨어공학과 교수
- 2025년 3월 ~ 현재 : 동신대학교 컴퓨터학과 박사과정

<관심분야>

사물인터넷, 스토리지 보안, 디지털 포렌식

류 갑 상(Gab-Sang Ryu)

[종신회원]



- 1985년 3월 ~ 1996년 2월 :
한국기계연구원, 선임연구원
- 1996년 3월 ~ 현재 : 동신대학교
컴퓨터학과 교수
- 2020년 1월 ~ 2021년 1월 :
한국소프트웨어품질안전포럼, 의장

〈관심분야〉

블록체인, SW품질, 정보처리