

논문 2018-13-31

연산 특성을 고려한 향상된 적응적 가비지 컬렉션 정책 (An Advanced Adaptive Garbage Collection Policy by Considering the Operation Characteristics)

박 송 화*, 이 정 훈, 이 원 오, 김 현 우

(Song-Hwa Park, Jung-Hoon Lee, Won-Oh Lee, Hyun-Woo Kim)

Abstract : NAND flash memory has widely been used because of non-volatility, low power consumption and fast access time. However, it suffers from inability to provide update-in-place and the erase cycle is limited. The unit of read/write operation is a page and the unit of erase operation is a block. Moreover erase operation is slower than other operations. We proposed the Adaptive Garbage Collection (called "AGC") policy which focuses on not only reducing garbage collection process time for real-time guarantee but also wear-leveling for a flash memory lifetime. The AGC performs better than Cost-benefit policy and Greedy policy. But the AGC does not consider the operation characteristics. So we proposed the Advanced Adaptive Garbage Collection (called "A-AGC") policy which considers the page write operation count and block erase operation count. The A-AGC reduces the write operations by considering the data update frequency and update data size. Also, it reduces the erase operations by considering the file fragmentation. We implemented the A-AGC policy and measured the performance compared with the AGC policy. Simulation results show that the A-AGC policy performs better than AGC, specially for append operation.

Keywords : NAND flash memory, Garbage collection, Wear-leveling

1. 서 론

플래시 메모리는 일종의 EEPROM (Electrically Erasable and Programmable ROM)으로 전기적으로 데이터를 지우고 다시 기록할 수 있는 비휘발성 (non-volatile) 기억 장치이다. 부피가 작고 가벼우며 소비 전력이 낮고 입출력 속도가 빠를 뿐만 아니라 EEPROM보다 가격이 싼 장점으로 인하여 저장 매체로서 사용이 증가하고 있다.

플래시 메모리를 저장장치로 사용하기 위해서는 극복해야 할 두 가지 단점이 있다. 첫째, 플래시 메모리는 데이터 수정 시 본래 주소에 덮어쓰기 (update-in-place)가 불가능하다. 플래시 메모리의

각 비트가 단방향으로만 토글링 (toggling)되기 때문에 쓰기 연산 시 지움 연산을 선행해야 한다. 즉, 쓰기 연산 전에 초기화된 메모리 공간의 확보가 필요하다. 둘째, 플래시 메모리의 각 블록은 지움 연산의 횟수가 제한되어 있기 때문에 플래시 메모리의 전체 공간이 균등하게 사용되지 못하면 사용가능한 메모리 공간이 급격하게 줄어들게 된다 [1].

이러한 플래시 메모리의 단점을 극복하고 임베디드 시스템에서의 저장장치로 사용하기 위하여 많은 연구가 진행되고 있다. Greedy 정책 [2, 3]은 유효하지 않은 자료가 가장 많은 블록을 선택하여 지움 연산을 수행하는 방법으로서, 삭제 횟수는 가능한 최소화하면서 많은 공간을 확보한다. 지움 연산의 효율성은 좋으나 블록의 지움 횟수가 고려되지 않아 플래시 메모리의 수명을 단축시킬 수 있는 단점이 있다.

Cost-benefit 정책 [4]은 특정 블록에 집중된 지움 연산으로 인하여 플래시 메모리의 수명이 단축되는 문제점을 개선하기 위하여 연구된 정책이다.

*Corresponding Author (shpark25@lignex1.com)

Received: July 31, 2018, Revised: Oct. 1, 2018,

Accepted: Oct. 2, 2018.

S. Park, J. Lee, W. Lee, H. Kim: LIG Nex1

표 1. 블록 상태 정보
Table 1. Block status information

Item	Description
Block Number	<ul style="list-style-type: none"> Based on the physical location of the block Sequential increase from 1
Degree of Invalidation	<ul style="list-style-type: none"> The sum of the number of empty pages and the number of invalid pages in the block
Erase Count	<ul style="list-style-type: none"> Block erase operation count
Erase Deviation	<ul style="list-style-type: none"> A value obtained by subtracting the erase count of block from the erase count average of the entire blocks
Score	<ul style="list-style-type: none"> Use this item to select erase(victim) blocks Score = Invalidity×W₁ + ED×W₂, (W₁ + W₂ = 1.0) <ul style="list-style-type: none"> - Invalidity : Degree of Invalidation - W₁ : Weight of the Invalidity - ED : Erase Deviation - W₂ : Weight of the ED The greater the efficiency of erase operations and wear-leveling effect, the higher the score

Cost-benefit 정책은 삭제의 효율성뿐만 아니라 특정 블록에 대한 지움 연산 집중을 피하기 위해, 블록이 사용된 최근 시간을 고려하여 삭제 할 블록을 선택한다.

콜드 블록 추적 (TCB : Tracking Cold Blocks) 기법 [5]은 콜드 블록의 인지수준을 강화하고 마모도 평균화로 인한 페이지들의 이주를 최소화함으로써 오버헤드를 감소시키고 수명을 증가시켜 플래시 메모리의 전반적인 성능을 향상시켰으나 블록 간 마모도 편차에 대한 보완이 필요하다.

적응적 가비지 컬렉션 (AGC: Adaptive Garbage Collection) 정책 [6]은 임베디드 시스템에서 실시간성을 보장하고 효율적인 지움 연산 및 지움 균등을 수행하기 위하여 시스템의 CPU 사용률에 따라 지움 연산 속도와 지움 균등을 고려한 가비지 컬렉션 기법이다. AGC 정책은 Greedy 정책에 비해 지움 균등 성능을 향상시키고, Cost-benefit 정책에 비해 연산 속도를 향상시켰으나 연산의 특성을 고려하지 못하였다. 이에 본 논문에서는 연산 특성을 고려하여 쓰기 연산 및 지우기 연산을 감소시킴으로서 성능을 향상한 AGC 정책을 제안한다.

본 논문의 구성은 다음과 같다. II장에서는 적응적 가비지 컬렉션 정책 및 파일 연산 특성 분석 결과에 대해 기술하고, III장에서는 본 논문에서 제안하는 향상된 적응적 가비지 컬렉션 정책에 대해 설명한다. IV장에서는 제안하는 정책의 성능을 실험을 통해 평가하고, V장에서는 결론 및 향후 과제에 대해 제시한다.

II. 관련 연구

1. 적응적 가비지 컬렉션 정책 [6]

적응적 가비지 컬렉션 (AGC) 정책은 실시간성을 보장하고 효율적인 지움 연산 및 지움 균등을 수행하기 위하여 블록 상태 정보, 페이지 상태 정보 및 자유 블록 정보를 관리하고 CPU 사용률에 따라 가비지 컬렉션 동작 모드를 설정한다.

1.1 플래시 메모리 상태 관리

적응적 가비지 컬렉션 정책은 지움 연산의 효율성과 균등 지움을 고려하기 위하여 블록 상태 정보, 페이지 상태 정보 및 자유 블록 정보를 관리한다.

지움 블록 선택 시, 지움 연산 속도와 균등 지움을 고려하기 위하여 표 1과 같이 플래시 메모리를 구성하는 블록의 상태 정보를 관리한다. 블록 무효도는 블록에 포함된 유효 페이지의 개수와 반비례하므로, 블록 무효도가 높은 블록일수록 지움 연산에 소요되는 시간이 적다. 블록 Score를 사용하여 지움 블록을 선택하는 경우, 무효도를 사용하여 지움 블록을 선택한 경우보다 지움 연산 속도는 느리지만 균등 지움 효과가 발생한다.

블록을 구성하는 페이지들의 갱신 경향이 유사할수록 해당 블록의 지움 연산을 효율적으로 처리할 수 있다. 적응적 가비지 컬렉션 정책에서는 페이지의 갱신 정도를 관리 및 예측하기 위하여 그림 1과 같이 페이지의 Age 정보를 관리한다.

		Page									
		1	2	3	4	5	6	...	N		
Block	1	60	45	10	30	20	32	...	20		
	2	5	4	3	0	9	0	...	0		
	3	5	11	7	20	3	5	...	0		
	M	7	11	8	9	30	30	...	100		

그림 1. 페이지 상태 정보의 예
Fig. 1 Example of page status information

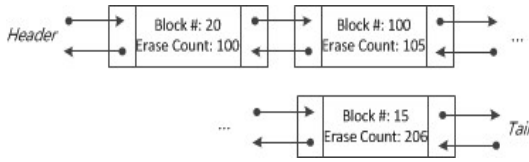


그림 2. 자유 블록 정보 이중 연결 리스트의 예
Fig. 2 Example of a doubly linked list to store the free block information

AGC 정책은 블록 할당을 빠르게 수행하기 위하여 그림 2와 같이 초기화된 자유 블록을 이중 연결 리스트로 관리한다. 리스트를 구성하는 노드는 블록의 번호와 지움 횟수 정보로 구성되며, 지움 횟수를 기준으로 오름차순으로 정렬된다. 지움 균등을 고려한 쓰기 블록 할당 시에는 리스트의 헤더 (header)가 가리키는 블록을 할당한다. 지움 연산 수행 중 Age가 낮은 페이지들을 백업하기 위하여 블록을 할당할 경우에는 테일 (tail)이 가리키는 블록을 할당한다.

1.2 가비지 컬렉션의 동작 모드

적응적 가비지 컬렉션 정책은 그림 3과 같이 CPU 사용률에 따라 가비지 컬렉션 동작 모드를 결정하며, 각 컬렉션 동작 모드는 표 2와 같은 특성을 가진다.

Fast Mode는 CPU 사용률이 높은 상황에서의 동작 모드로 CPU 사용률의 최소화과 빠른 플래시 처리 응답을 위하여 지움 연산의 부하가 최소화되도록 동작한다. 이를 위하여 Fast Mode는 가비지 컬렉션 수행 시, 무효도가 높은 블록을 지움 블록으로 선택한다.

Smart Mode는 지움 연산의 효율성과 균등 지움을 동시에 고려한 모드로, 적응적 가비지 컬렉션의 기본 모드이다. Smart Mode는 지움 블록 선택 시, 블록의 무효도와 지움 횟수를 고려하기 위하여 블록의 Score 정보를 사용하며, Score가 높은 블록을 지움 블록으로 선택한다.

Wear-leveling Mode의 기본 동작은 Smart Mode와 동일하며, 다른 모드와 달리 유효 페이지 복사 이전에 유효 페이지의 재그룹화 과정을 수행하여 유효 페이지 복사 과정에서 지움 균등을 고려한다. Wear-leveling Mode는 유효 페이지 복사 전에 지움 블록으로 선택된 블록들의 유효 페이지들을 페이지 상태 정보를 사용하여 Age가 유사한 페이지들을 그룹화 한다. 지움 횟수가 낮은 블록에 갱신 빈도가 높은 데이터를 저장하며, 지움 횟수가

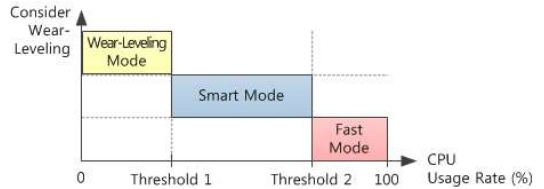


그림 3. CPU 사용률에 따른 적응적 가비지 컬렉션 동작 모드

Fig. 3 Operation mode of adaptive garbage collection on CPU usage rate

표 2. 적응적 가비지 컬렉션의 동작 모드

Table 2. Operation mode of the adaptive garbage collection

Category		Fast Mode	Smart Mode	Wear-leveling Mode
CPU Usage		High	Medium	Low
Consider erase operation speed		O	O	O
Consider Wear-leveling	Selection of an erase block	X	O	O
	Select of a backup block	X	X	O

높은 블록에 갱신 빈도가 낮은 데이터를 저장함으로써 균등 지움이 발생되도록 한다.

2. 파일 연산 특성 분석

로그 기록을 위한 파일을 생성하는 경우, 해당 파일에 대해서는 파일 끝에 추가 쓰기 (append)가 빈번하게 발생한다. 반면 시스템 파일의 경우에는 업데이트가 거의 발생하지 않거나, 파라미터 갱신을 위하여 일정 부분에 대해서만 업데이트가 발생한다.

NAND 플래시 메모리를 저장 장치로 사용할 경우, 데이터 쓰기는 페이지 단위로 발생한다. 따라서 파일에 추가 쓰기가 빈번하게 발생하는 경우, 그림 4와 같이 실제 데이터 크기가 아닌 페이지 단위의 쓰기 연산이 발생한다. 이런 경우, 실제 플래시 메모리에 기록되는 데이터 크기에 비하여 플래시 메모리 쓰기 및 지움 연산이 자주 발생하여 플래시 메모리 연산 속도가 느려지고 수명이 빠르게 감소될 수 있다.

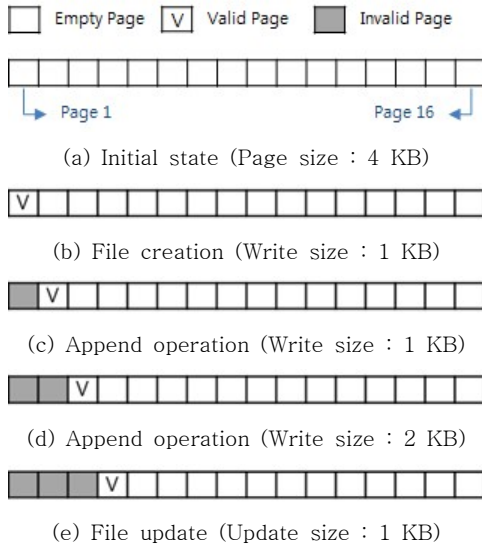


그림 4. 파일 연산 특성에 따른 플래시 메모리 상태
Fig. 4 Flash memory status on file operation characteristics

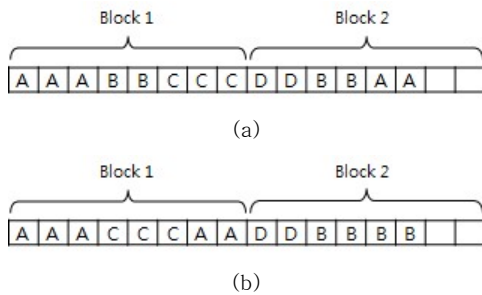


그림 5. 파일 단편화에 따른 플래시 메모리 상태
Fig. 5 Flash memory status on file fragmentation

파일의 단편화 정도는 파일 삭제의 성능에 영향을 미친다. 그림 5의 (a)에서 파일 A 삭제 시에는 블록 1과 블록 2에서 각각 3 개, 2 개의 페이지가 무효화 처리된다. 또한 지움 연산이 발생할 경우, 각각 5 개, 4 개의 페이지에 대해 읽기 및 쓰기가 발생한다. 반면 그림 5의 (b)에서 파일 A 삭제 시에는 블록 1에서 5 개의 페이지가 무효화 처리된다. 또한 지움 연산이 발생할 경우, 3 개의 페이지에 대해 읽기 및 쓰기가 발생한다. 즉, 파일의 단편화가 적을수록 파일 삭제 연산을 효율적으로 처리할 수 있다.

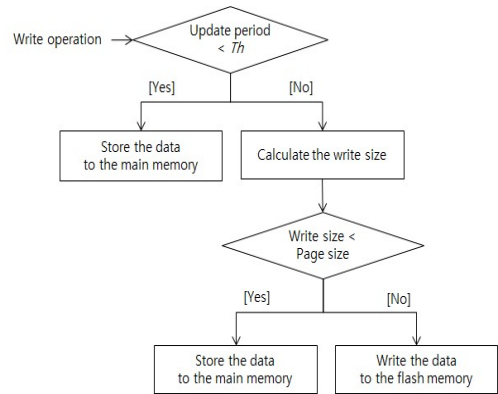


그림 6. A-AGC 정책의 쓰기 연산 처리 과정
Fig. 6 Write operation process of A-AGC policy

이에 본 논문에서는 쓰기 연산 및 파일 단편화를 감소시킴으로써 플래시 연산 속도를 향상하고 플래시 메모리의 수명을 증가시킬 수 있는 향상된 적응적 가비지 컬렉션 정책을 제안하고자 한다.

III. 향상된 적응적 가비지 컬렉션 정책

본 절에서는 기존에 제안한 적응적 가비지 컬렉션 정책에 연산 특성을 고려하여 쓰기 연산 및 파일 단편화를 감소시킴으로써 성능이 향상된 적응적 가비지 컬렉션 (A-AGC : Advanced-Adaptive Garbage Collection) 정책에 대하여 기술한다.

1. 쓰기 연산

향상된 적응적 가비지 컬렉션 정책에서는 성능 향상을 위하여 플래시 메모리 쓰기 횟수를 감소시킨다. A-AGC 정책의 쓰기 연산 처리 과정은 그림 6과 같다.

A-AGC 기법에서는 데이터의 갱신 빈도 및 쓰기 데이터의 크기를 고려하여 쓰기 연산을 수행한다. 데이터의 갱신 빈도가 높은 경우에는 플래시 메모리에 쓰지 않고 버퍼에 저장한다. 데이터의 갱신 빈도가 낮은 경우에는 데이터를 플래시 메모리에 쓰지만 갱신 빈도가 높거나, 갱신 빈도가 높더라도 데이터의 크기가 페이지보다 작은 경우에는 데이터를 버퍼에 저장한다. 단, 파일 별로 버퍼링된 데이터의 크기가 페이지보다 큰 경우에는 페이지 단위로 데이터를 플래시 메모리에 기록한다. 이를 위해서 A-AGC에서는 파일 별 버퍼, 데이터의 위치 및 크기 등을 관리한다.

표 3. 파일 객체 정보의 관리

Table 3. Management of the file object information

Item	Description
Object ID	• Identifier of the file
Object Size	• Size of the file
Data Status	<ul style="list-style-type: none"> • Data storage status • The status of one of the following : <ul style="list-style-type: none"> - NO DATA : No data is written - WRITTEN : Data is written at the flash memory - BUFFERED : Data is stored at the buffer (RAM) - INVALID : Data written at the flash memory is deleted
Data Location	• An array of physical page numbers where data is stored in flash memory
Data Size	• Size of data stored in flash memory or buffer
File Update Counter	<ul style="list-style-type: none"> • Number of times the file was updated • The number of updates is 1 at the time of initial write, and is incremented by 1 for each update
Buffer Address	• If the data status is BUFFERED, the address of the main memory where the data is stored

표 3은 향상된 적응적 가비지 컬렉션 정책에서 관리하는 파일 객체의 정보를 나타낸다. 향상된 적응적 가비지 컬렉션 정책은 파일 생성 시, 파일 정보 관리를 위한 파일 객체 정보를 생성한다. 각 파일의 데이터는 플래시 메모리에 저장되는 페이지 단위로 관리하며, 페이지 단위의 데이터 별로 저장되는 물리적 위치 및 상태 정보 등을 관리한다. 파일에 대한 읽기/수정/삭제 등의 연산 발생 시, 해당 객체 정보에 저장된 데이터 저장 위치, 크기 등을 사용하여 파일 연산을 수행한다.

그림 7은 그림 4의 파일 연산 수행을 향상된 적응적 가비지 컬렉션 정책을 사용한 경우를 나타낸다. 향상된 적응적 가비지 컬렉션 정책을 적용한 경우, 기존 정책에 비하여 플래시 메모리 쓰기 연산 횟수가 감소되는 것을 확인할 수 있다. 쓰기 연산 횟수의 감소는 무효 페이지의 감소, 빈 블록 개수의 증가 효과가 있기 때문에 궁극적으로는 지움 연산 횟수가 감소되는 효과가 있다.

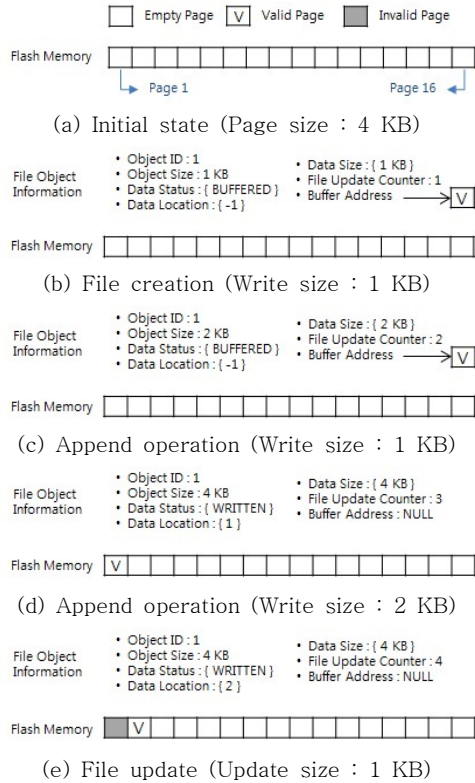


그림 7. 향상된 적응적 가비지 컬렉션 정책에서의 파일 쓰기

Fig. 7 File write operation of advanced adaptive garbage collection policy

2. 가비지 컬렉션의 동작

향상된 적응적 가비지 컬렉션 정책은 기존의 적응적 가비지 컬렉션 정책과 동일하게 시스템의 CPU 사용률에 따라 Fast Mode, Smart Mode, Wear-leveling Mode 중 하나의 모드로 동작한다. 향상된 적응적 가비지 컬렉션 정책은 파일 단편화의 감소를 위하여 가비지 컬렉션 수행 시, 동일 파일의 데이터들이 가능한 물리적으로 연속된 위치에 배치되도록 유효 페이지의 재그룹화 과정을 수행한다.

2.1 Fast Mode

Fast Mode는 CPU 사용률이 높은 상황에서의 동작 모드로, 향상된 적응적 가비지 컬렉션 정책의 Fast Mode 동작은 기존 정책의 Fast Mode 동작과 동일하다. Fast Mode는 가비지 컬렉션 수행 시, 무효도가 높은 블록을 지움 블록으로 선택한다.

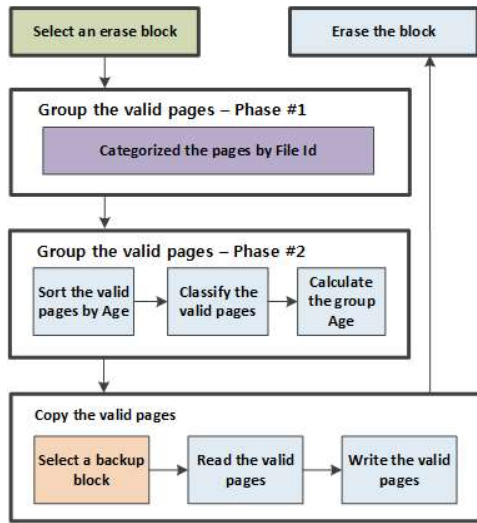


그림 8. Wear-leveling 모드 동작
Fig. 8 Wear-leveling mode operation

2.2 Smart Mode

Smart Mode는 지움 연산의 효율성과 균등 지움을 동시에 고려한 모드로, 향상된 적응적 가비지 컬렉션 정책의 Smart Mode 동작은 기존 정책의 Smart Mode 동작과 동일하다. Smart Mode는 지움 블록 선택 시, 블록의 무효도와 지움 횟수를 고려하기 위하여 블록의 Score 정보를 사용하며, Score가 높은 블록을 지움 블록으로 선택한다.

2.3 Wear-leveling Mode

Wear-leveling Mode의 기본 동작은 Smart Mode와 동일하며, 다른 모드와 달리 유효 페이지 복사 이전에 유효 페이지의 재그룹화 과정을 수행하여 유효 페이지 복사 과정에서 지움 균등을 고려한다.

기존 정책에서의 Wear-leveling Mode는 유효 페이지의 재그룹화 시에 Age가 유사한 페이지들을 동일 블록에 저장하였다. 이때, 갱신 빈도가 높아 Age가 낮은 페이지들은 지움 횟수가 낮은 블록에 저장하였으며, 갱신 빈도가 낮아 Age가 높은 페이지들은 지움 횟수가 높은 블록에 저장하였다.

향상된 적응적 가비지 컬렉션 정책에서는 기존 정책에서의 Wear-leveling Mode와 동일하되, 그림 8과 같이 파일 단편화 감소를 위하여 유효 페이지들의 재그룹화 과정을 2 단계로 수행한다. 파일 별로 1차 그룹화를 수행하고, 동일 그룹 내에서 Age가 유사한 페이지들에 대해 2차 그룹화를 수행한다.

표 4. 시뮬레이션 파라미터
Table 4. Parameter of simulation

Item	Value
#blocks/chip	2,048
#pages/block	64
Page Read/Write Latency	60 us / 800 us
Block Erase Latency	1.5 ms
Program-Erase (P/E) Cycle	10,000

표 5. 제안한 가비지 컬렉션 정책의 파라미터
Table 5. Parameter of proposed garbage collection policy

Category	Item	Value
Operation Mode	Threshold ₁	30.0
	Threshold ₂	70.0
Score	W ₁	0.5
	W ₂	0.5

1, 2차 그룹화 결과에 따라 그룹화 된 페이지들을 동일 블록에 저장한다. 또한 버퍼링된 데이터의 소실을 대비하여 버퍼링된 데이터를 플래시 메모리에 저장한다.

향상된 적응적 가비지 컬렉션 정책에서는 파일 단편화를 감소시킴으로서 파일 삭제 및 갱신으로 인하여 발생할 수 있는 블록 지움 연산의 비용을 최소화시킬 수 있다.

IV. 성능 평가

1. 실험 환경

본 논문에서 제안한 적응적 가비지 컬렉션 정책의 성능을 평가하기 위하여 SSD Extension for DiskSim Simulation을 참고하였으며 [7, 8], 시뮬레이션 환경은 표 4와 같다.

제안한 기법에 적용한 파라미터 값은 표 5와 같으며, 실험에 사용된 시스템의 CPU 사용률과 제안한 가비지 컬렉션의 동작 모드는 그림 9와 같다.

2. 기존 기법과의 성능 비교

향상된 적응적 가비지 컬렉션 (A-AGC) 정책의 성능을 평가하기 위하여 기존의 적응적 가비지 컬

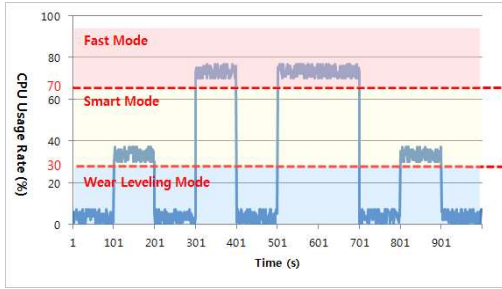


그림 9. 시험에 사용된 CPU 사용률 및 가비지 컬렉션의 동작 모드

Fig. 9 CPU usage rate and operation mode of proposed garbage collection for the test

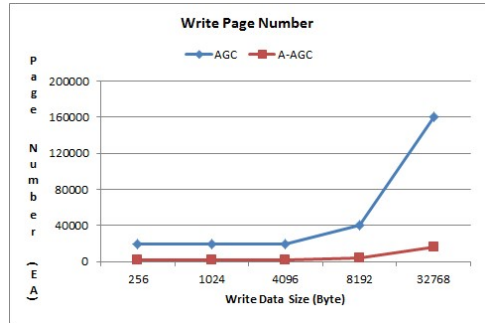


그림 10. 추가 쓰기 데이터 크기에 따른 쓰기 페이지 개수 비교

Fig. 10 Comparison of the written page number according to the append data size

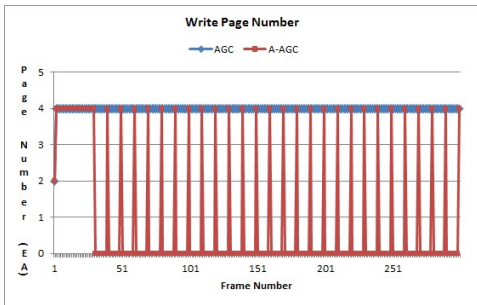


그림 11. 추가 쓰기 데이터의 크기가 8 KB 인 경우의 쓰기 페이지 개수 비교

Fig. 11 Comparison of the written page number when the append data size is 8 KB

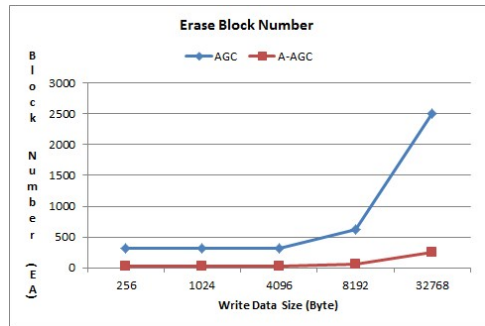


그림 12. 지움 블록 개수 비교

Fig. 12 Comparison of the erased block number

렉션 (AGC) 정책의 성능과 비교하였다. 두 가지 기법에 대해 동일한 환경에서 동일한 파일 연산을 수행하도록 하였으며 가비지 컬렉션의 수행 시점도 동일하도록 하였다.

2.1 추가 쓰기 연산 성능의 비교

추가 쓰기 연산 시의 성능을 비교하기 위하여 표 6의 연산이 발생하도록 하고, 쓰기 연산이 발생한 페이지 개수와 지움 연산이 발생한 블록 개수를 측정하였다.

그림 10은 추가 쓰기 데이터 크기에 따른 쓰기 페이지 개수를 나타내고 있으며, 제안 기법이 기존의 기법에 비하여 쓰기 연산을 위한 페이지를 적게 사용하였다.

그림 11은 추가 쓰기 데이터의 크기가 8 KB 인 경우의 쓰기 페이지 개수를 나타내고 있으며, 제안

기법이 기존 기법에 비하여 쓰기 연산을 위한 페이지를 약 90 % 적게 사용하였다. 이는 쓰기 데이터의 크기가 페이지 크기보다 크고 갱신 빈도가 높은 경우에는 제안 기법이 데이터를 버퍼링하기 때문인 것으로 판단된다.

그림 12는 지움 블록 개수를 나타내고 있으며, 제안 기법을 사용한 경우에 기존 기법에 비하여 지움 블록이 약 90 % 적게 발생되었다. 지움 블록 개수는 플래시 메모리의 사용률에 따라 많이 발생하게 되므로 그림 10과 유사한 경향을 보임을 알 수 있다.

2.2 랜덤 연산 성능의 비교

랜덤 연산 시의 성능을 비교하기 위하여 표 7의 연산이 발생하도록 하고, 사용된 플래시 메모리 페이지 수와 블록에 발생한 지움 횟수를 측정하였다.

표 6. 실험에 사용된 파일 연산

Table 6. File operation used at experiment

No	Append data size	No	Append data size
1	256 Byte	4	8192 Byte (8 KB)
2	1024 Byte (1 KB)	5	32768 Byte (32 KB)
3	4096 Byte (4 KB)		

표 8. 랜덤 연산 시의 쓰기 페이지 개수 비교

Table 8. Comparison of the written page number on random operation

Method	Write Page Number
AGC	448,187
A-AGC	335,428

랜덤 연산 시에는 표 8, 9와 같이 쓰기 페이지 개수와 지움 블록 개수가 각각 25 %, 8 % 감소하였다. 추가 쓰기 연산과 비교하였을 때 성능 향상의 정도는 낮으나 추가 쓰기 연산이 아닌 경우에도 성능 향상이 있음을 확인할 수 있다.

파일 단편화가 작을수록 지움 블록 개수가 감소되는데 파일 단편화 감소를 위한 유효 페이지 재그룹화가 Wear-leveling 모드에서만 동작하기 때문에 성능 향상의 정도가 낮은 것으로 판단된다.

V. 결 론

본 논문에서는 기존에 제안했던 적응적 가비지 컬렉션 정책에서 플래시 메모리 쓰기 및 파일 단편화를 감소시킴으로서 성능을 향상한 정책을 제안하였다. 제안 기법은 갱신 빈도가 높거나 쓰기 데이터의 크기가 작은 경우를 고려하여 쓰기 연산을 수행함으로써 플래시 메모리 쓰기 연산 횟수를 감소시켰다. 또한 가비지 컬렉션 수행 시, 파일의 단편화를 감소시킴으로서 지움 블록 개수 및 지움 연산 비용이 감소되도록 하였다.

실험을 통하여 제안한 기법이 기존 기법 대비 성능이 향상되었음을 확인하였다. 특히 순차적 쓰기 연산이 발생하는 경우에는 약 90 %의 성능 향상을 보였다.

표 7. 실험에 사용된 파일 연산

Table 7. File operation used at experiment

File Operation	Description
Create	• File Size : 1 KByte ~ 4 MByte
Reate	• File Selection : Random
Update	• File Selection : Random • Update Location : Random • Update Size : Random
Delete	• File Selection : Random

표 9. 랜덤 연산 시의 지움 블록 개수 비교

Table 9. Comparison of the erased block number on random operation

Method	Erase Block Number
AGC	7,002
A-AGC	6,403

향후 본 연구는 순차적 쓰기 연산 외에 다양한 파일 연산 특성을 분석 및 고려하여 플래시 연산 성능 및 플래시 메모리의 수명 연장 등을 향상시킬 수 있는 방향으로 나아가야 할 것이다.

References

- [1] F. Dougli, R. Caceres, M.F. Kaashoek, P. Krishnan, K. Li, B. Marsh, J. Tauber, "Storage Alternative for Mobile Computers," Proceedings of the 1st USENIX Symposium on Operating System Design and Implementation, pp. 25-37, 1994.
- [2] M.L. Chiang, P.C.H. Lee, R.C. Chang, "Managing Flash Memory in Personal Communication Devices," Proceedings of IEEE Symposium on Consumer Electronics, pp. 177-182, 1997.
- [3] M.L. Chiang, R.C. Chang, "Cleaning Policies in Mobile Computers Using Flash Memory," Journal of System and Software, Vol. 48, No. 3, pp. 213-231, 1999.
- [4] A. Kawaguchi, S. Nishioka, H. Motoda "A Flash-Memory Based File System," Proceedings of 1995 USENIX Technical

- Conference, pp. 155–164, 1995.
- [5] Y.H. Jang, S.H. Kim, S.H. Hwang, M.S. Lee, C.H. Park, “Tracking Cold Blocks for Static Wear Leveling in FTL-based NAND Flash Memory,” IEMEK Journal of Embedded Systems and Applications, Vol. 12, No. 3, pp.185–192, 2017 (in Korean).
- [6] S.H. Park, J.H. Lee, W.O. Lee, H.E. Kim, “A Adaptive Garbage Collection Policy for Flash-Memory Storage System in Embedded System,” IEMEK Journal of Embedded Systems and Applications, Vol. 12, No. 3, pp.121–130, 2017 (in Korean).
- [7] V. Prabhakaran, T. Wobber, “SSD Extension for DiskSim Simulation Environment.” Microsoft Research 2009.
- [8] K9GAG08U0M 2G x 8bit MLC NAND Flash Memory Data Sheet, Samsung Electronics, <https://www.samsung.com>, 2007.

Song-Hwa Park (박 송 화)



Park received a B.S. and M.S. degrees in Computer Engineering from Pusan National University in 2005 and 2007, respectively. She is currently a research engineer at LIG Nex1. Her research interests include flash file system, embedded system and track fusion.
Email: shpark25@lignex1.com

Jung-Hoon Lee (이 정 훈)



Lee received a B.S., M.S. and Ph.D. degrees in Electronic Engineering from Kyungpook National University in 1997, 1999 and 2004, respectively. He is now a research engineer at LIG Nex1. His research interests include SONAR system, embedded system, and signal processing.
Email: junghoon.lee04@lignex1.com

Won-Oh Lee (이 원 오)



Lee received a B.S. and M.S. degrees in Electrical Engineering and Computer Science from Kyungpook National University in 2008 and 2011, respectively. Currently, he is a research engineer at LIG Nex1. His research interests include computer vision and pattern recognition in embedded system.
Email: wonoh.lee@lignex1.com

Hyun-Woo Kim (김 현 우)



Kim is received B.S. degree in electronic engineering from Korea University in 2018. He is now a research engineer at LIG Nex1. His research interests include embedded system and software engineering.
Email: hyunwoo.kim@lignex1.com