

NVMe Flexible Data Placement : Reporting Feature 에뮬레이션 연구

이름 이근호

지도교수 김상훈

연구배경

Flash memory에서 발생하는 Write Amplification(WA)은 스토리지의 성능 저하와 수명 단축의 주요 원인이다. 이를 해결하기 위해 2022년 11월에 제안된 Flexible Data Placement(FDP)는 데이터를 유연하게 배치하여 WA를 줄이는 기술로, Reclaim Unit(RU), Reclaim Group(RG), Reclaim Unit Handle(RUH) 등으로 구성된다. 각 구성 요소를 통해 데이터의 위치와 상태 정보를 관리하여 효율적인 데이터 배치를 가능하게 한다.

이전 연구에서는 FDP의 Read/Write와 I/O Management Receive(IOMR) 기능 에뮬레이션을 구현 및 검증하였다. 그러나 Garbage Collection(GC)을 포함한 핵심적인 데이터 관리 기능이 구현되지 않아, 완전한 시스템 동작을 확인하지는 못하였다.

본 연구는 FDP의 핵심 기능인 Reporting을 구현하는데 중점을 두고 있다. 이를 위해 I/O Management Send(IOMS), Get Log Page, Get/Set Features command 등을 구현하였으며, 데이터 배치 정보를 효과적으로 수집하고 WA 감소를 위한 기반을 마련하였다.

연구 진행 과정

Garbage Collection

1. Priority queue를 통해 victim RU 탐색
2. 상황에 따라 MBE, MBMW 증가 및 Media Reallocated 이벤트 생성
3. Victim RU의 valid page 복사
4. Victim RU를 free 상태로 전환

* Media Reallocated event : Victim RU의 valid page를 다른 곳에 복사하였음을 알리기 위함

I/O Management Send

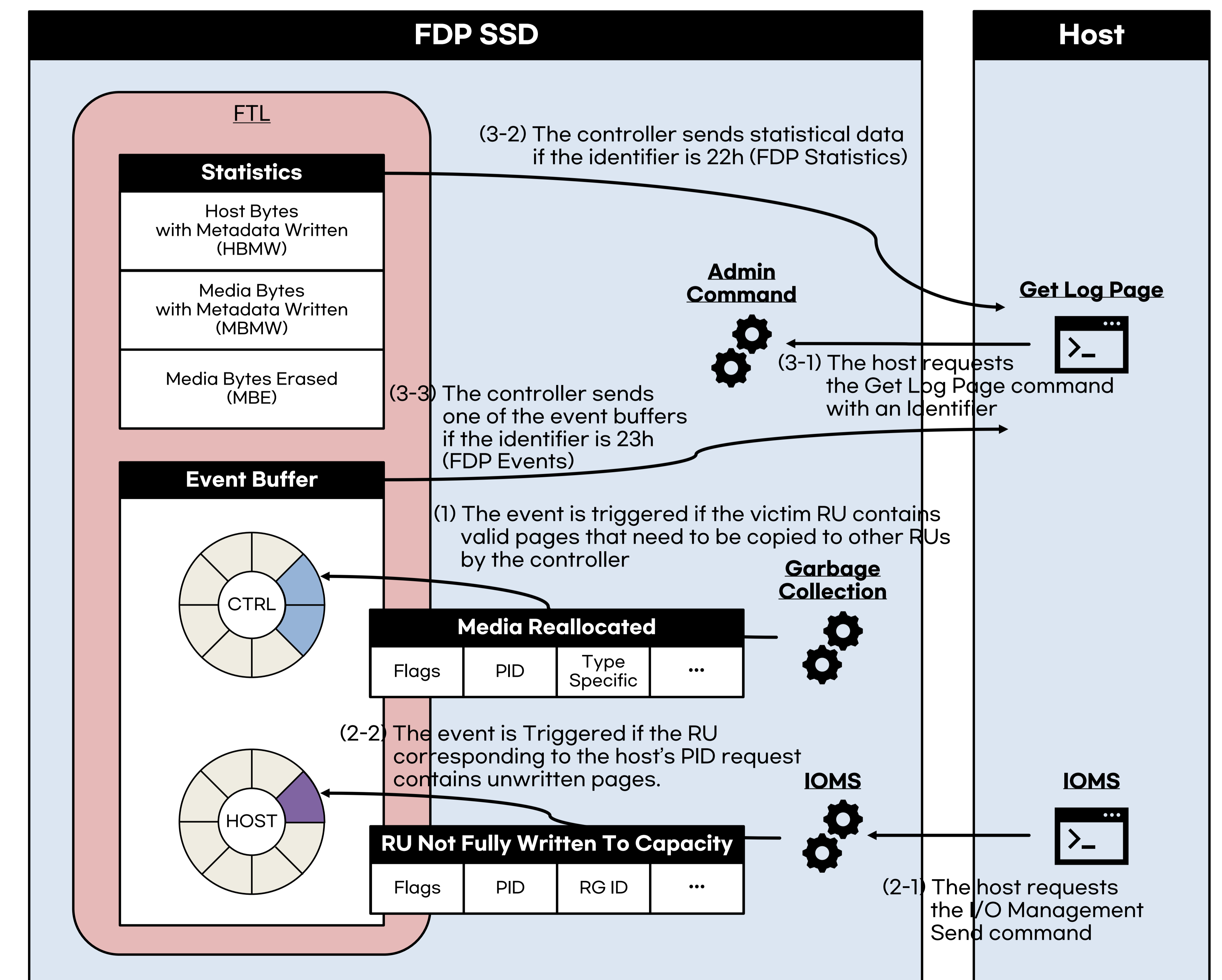
1. Host로부터 PID 확인
 2. PID에 대한 RUH의 Write Pointer(WP) 업데이트
- * RU Not Fully Written event : WP가 가리키는 RU에 valid page가 남아 있었음을 알리기 위함

Get Log Page

1. Host의 identifier에 따라 적절한 SSD 상태 값 전달
- * FDP Events (23h) : Command 설정에 따라 controller 혹은 host에 대한 event buffer 하나를 전달

Get/Set Features

1. Host의 identifier에 따라 적절한 상태 값 전달/설정
- * FDP Events (1Eh) : 특정 RUH에 대한 event filter 설정 가능



* Reclaim Unit

Page로 구성된, 관리 최소 단위. Valid/Invalid page 개수 관리.

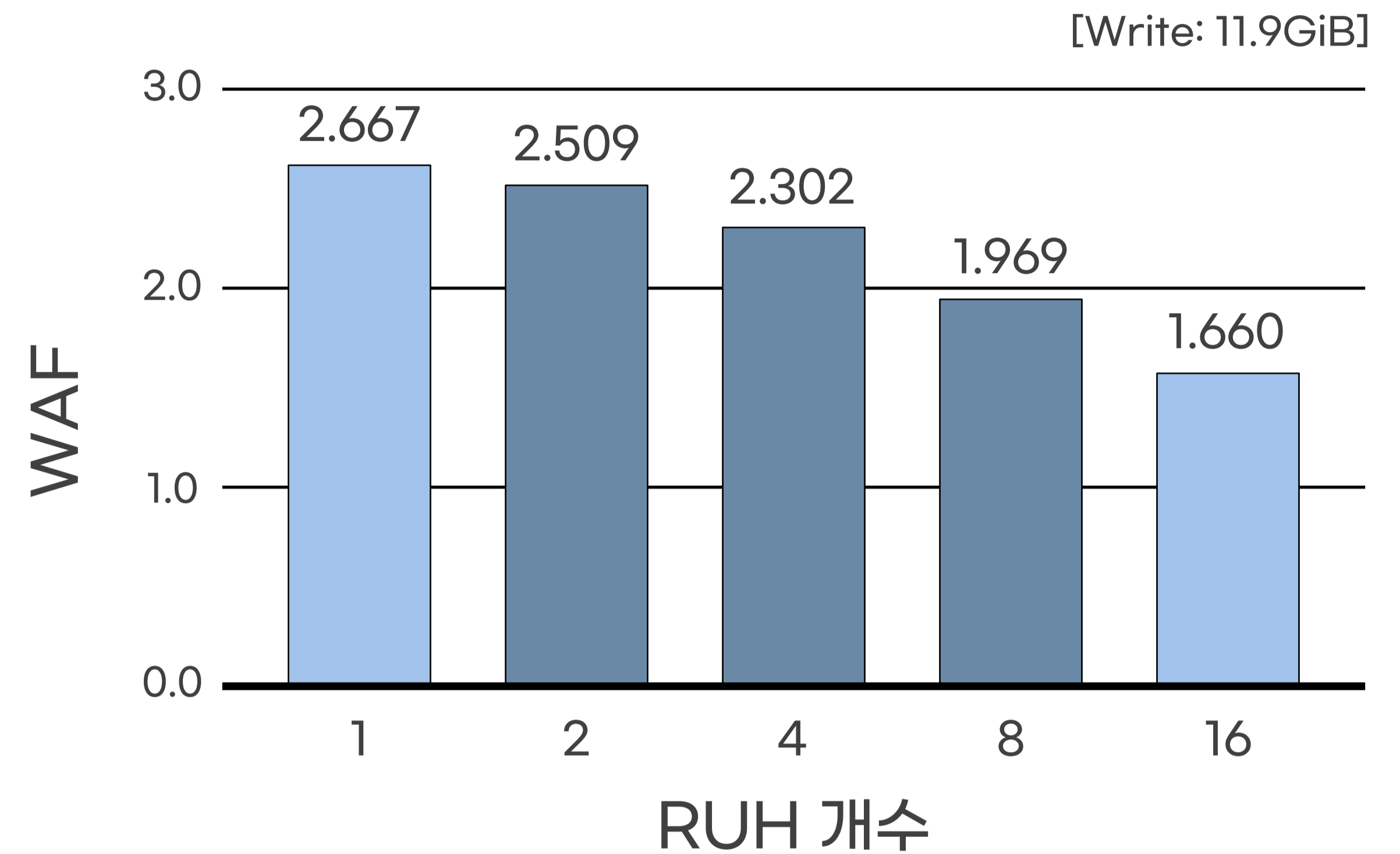
* Reclaim Group

RU로 구성된 관리 단위. GC를 위해 free 해주어야 하는 RU들 관리.

* Reclaim Unit Handle

각 RG에 속하는 RU를 가리킴. 특정 RU에 대해 다음 쓰기 위치 정보를 가짐.

결과 및 분석



위 그래프는 각 경우에 대해 11.9GiB 쓰기를 수행한 결과이다. RUH를 하나만 사용하는 경우는 FDP가 아닌 일반 SSD와 동일한 쓰기 방식을 사용하며, 이때 2.667의 WAF 수치를 보인다. 즉 호스트 요청보다 2.7배 가량 더 많은 데이터가 쓰이면서 수명이 단축된다. 이 수치는 16개에 비해 요청을 한 번 더 수행하는 것과 동일한 모습이다.

* WA 현상

호스트가 실제로 쓰기 요청한 용량보다 더 많은 양이 저장 장치에 쓰이는 현상

* WAF

WA 현상의 수치화. 실제로 쓰인 용량과 호스트 요청 용량 간의 비율을 계산.

오픈소스 URL

- NVMeVirt

<https://github.com/snu-csl/nvmevirt>

