

IDFTL (Indexing DFTL)

2024.08.28

Presentation by Dayeon Wee

wida10@dankook.ac.kr

Contents

1. Motivation
2. Evaluation
3. Plan

Motivation

gc_write_page

: GC 후 valid page를 어디에 write 할지에 대한 함수

특정 위치에 새로운 ppa 갱신

어디에 write 할지 포인터 옮기기

```
/* move valid page data (already in DRAM) from victim
static uint64_t gc_write_page(struct ssd *ssd, struct
{
    struct ppa new_ppa;
    struct nand_lun *new_lun;
    uint64_t lpn = get_rmap_ent(ssd, old_ppa);

    ftl_assert(valid_lpn(ssd, lpn));
    new_ppa = get_new_page(ssd);
    /* update maptbl */
    set_maptbl_ent(ssd, lpn, &new_ppa);
    /* update rmap */
    set_rmap_ent(ssd, lpn, &new_ppa);

    mark_page_valid(ssd, &new_ppa);

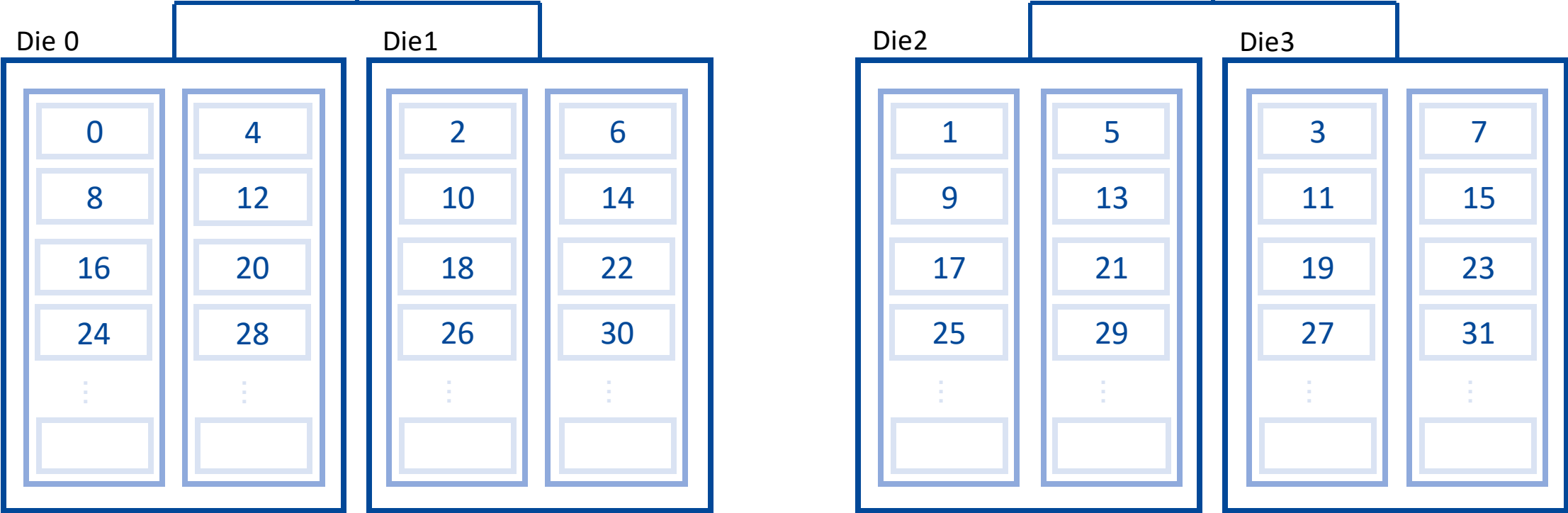
    /* need to advance the write pointer here */
    ssd_advance_write_pointer(ssd);

    if (ssd->sp.enable_gc_delay) {
        struct nand_cmd gcw;
        gcw.type = GC_IO;
        gcw.cmd = NAND_WRITE;
        gcw.stime = 0;
        ssd_advance_status(ssd, &new_ppa, &gcw);
    }
}
```

Motivation

Overview

- > 초기 페이지 할당 방식
- > GC 이후 페이지 작성 방식



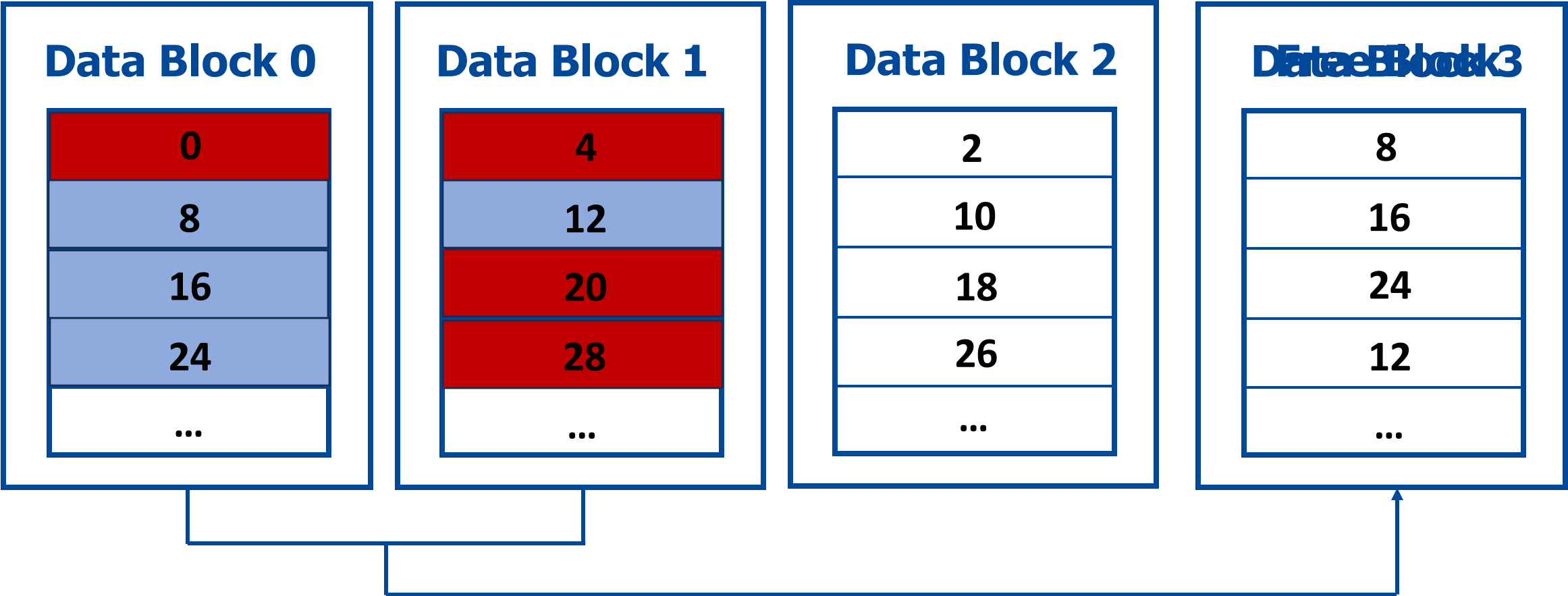
Motivation

GC Threshold ↑

GC Not yet !

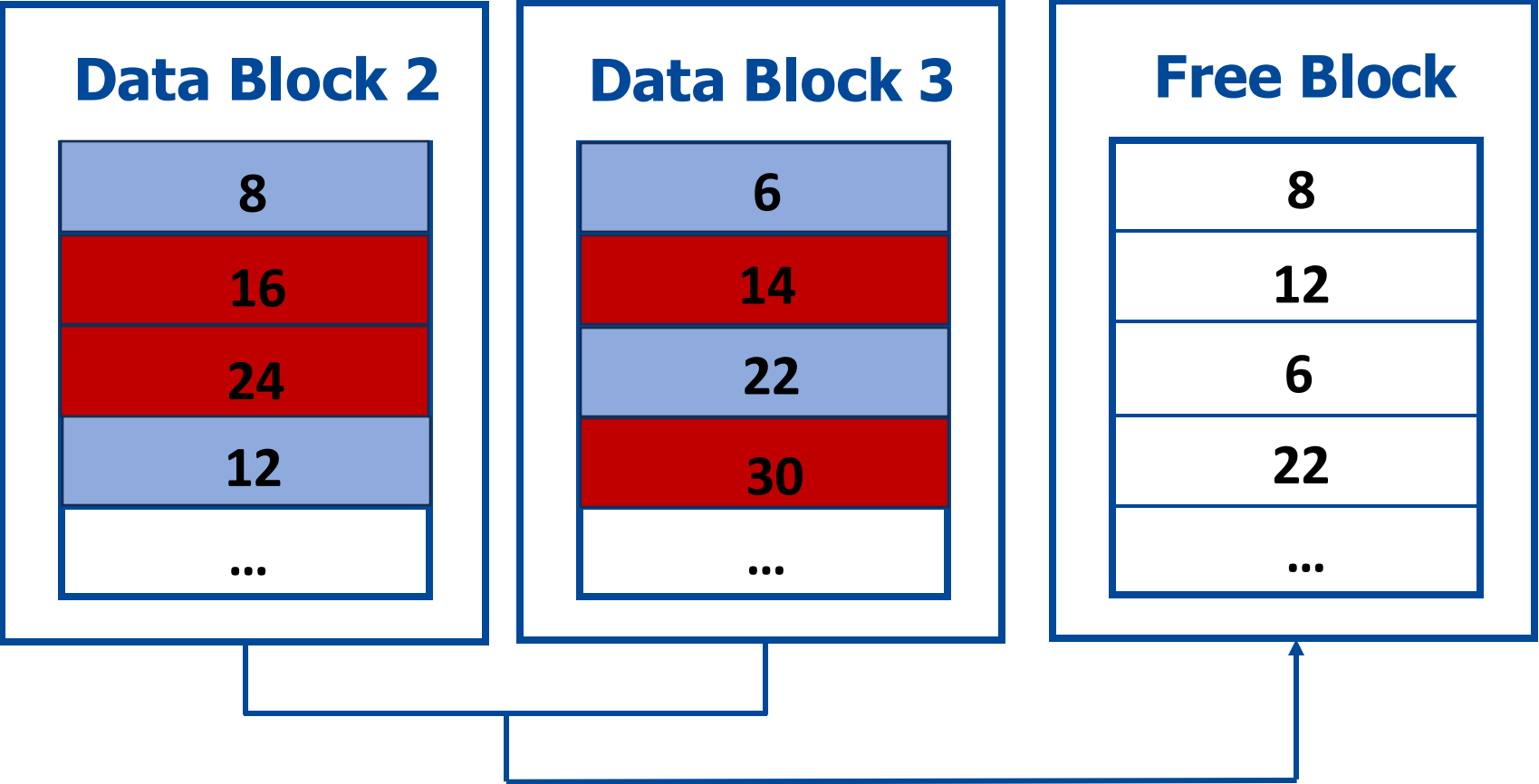
: invalid page

: valid page



Motivation

GC Threshold ↑



기존 방식
: 한 블록에서 valid page 다 넣고
그 다음 블록에서 valid page 넣음

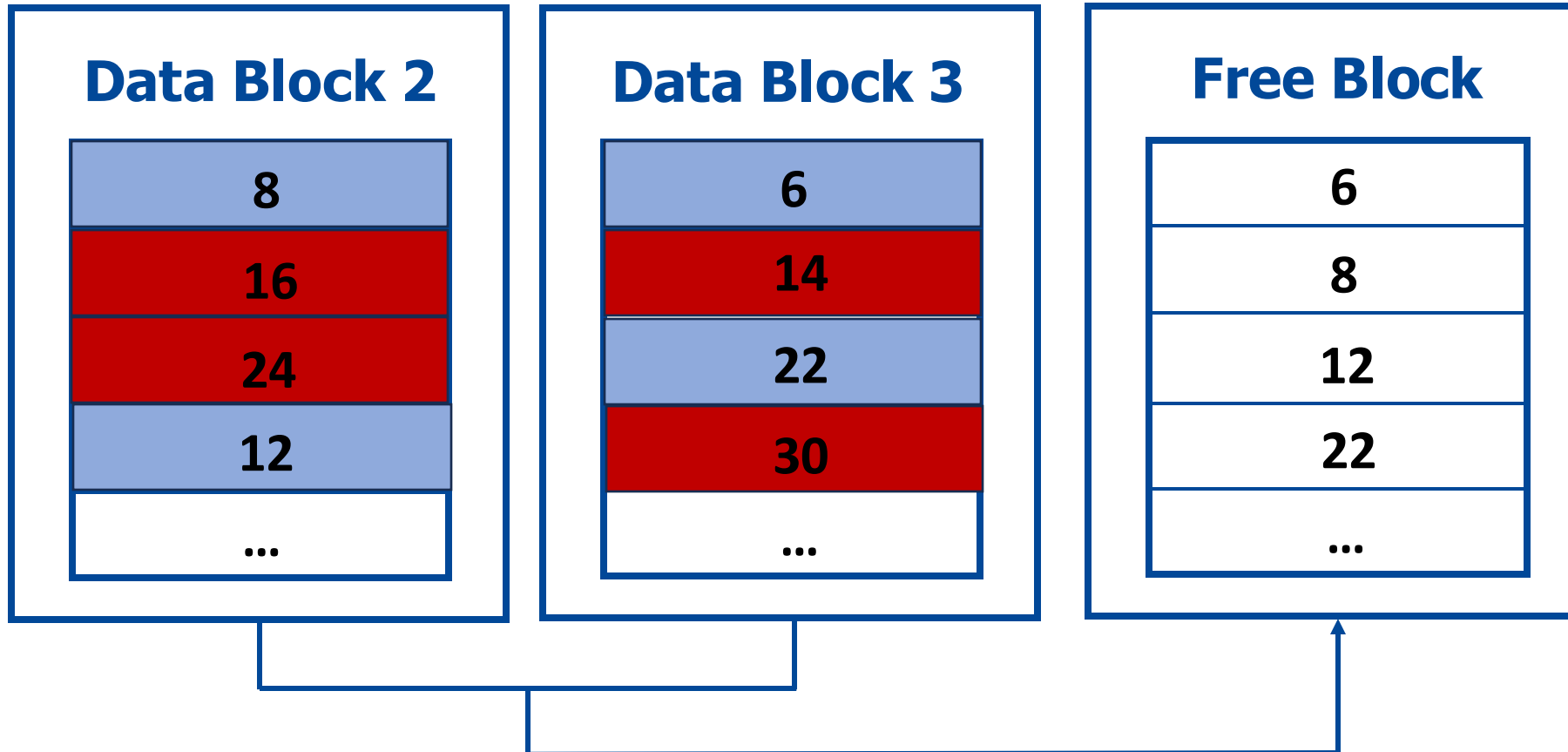
↓
이러면 섞이지 않을까?

↓
인덱싱으로 GC해도 섞이지않게
정렬되게끔 해보자!
(지역성 활용 !)

Motivation

가까운 인덱스끼리 모여져있게끔..

LPN이나 인덱싱이나 같은 수지만,,, LPN으로 정렬해볼까 했지만 구현 아직까진 실패..



Evaluation

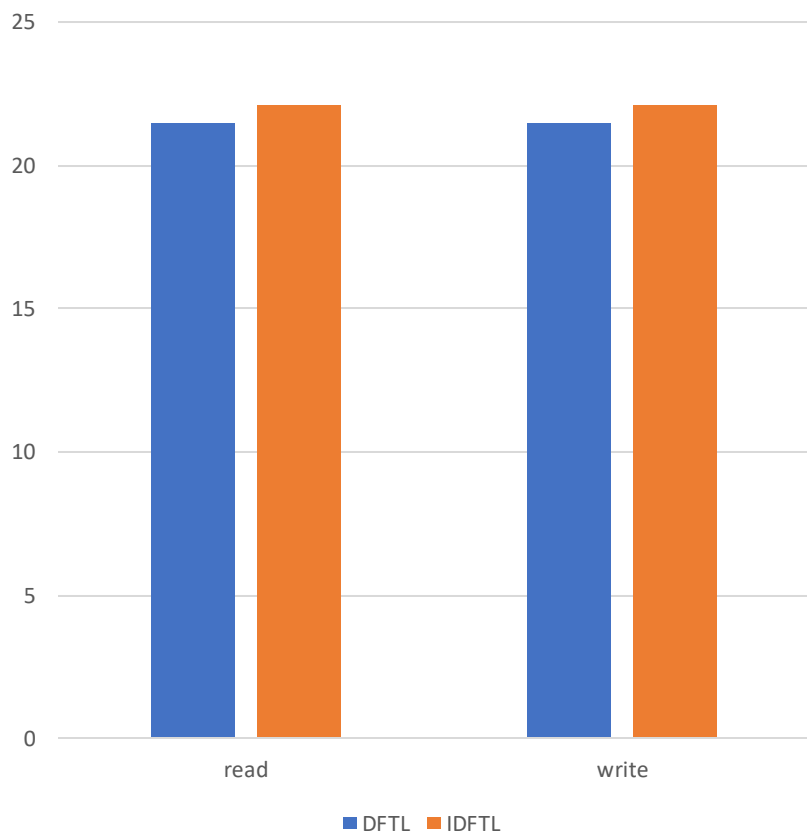
FIO (random read / write)

항목	값
블록 크기	4 KB
파일 크기	2 GB
캐시 사용여부	사용 안함 (O_DIRECT 사용)
작업 수행 시간	180 s
병렬 작업 수	4
I/O engine	libaio
RW Ratio	50 : 50

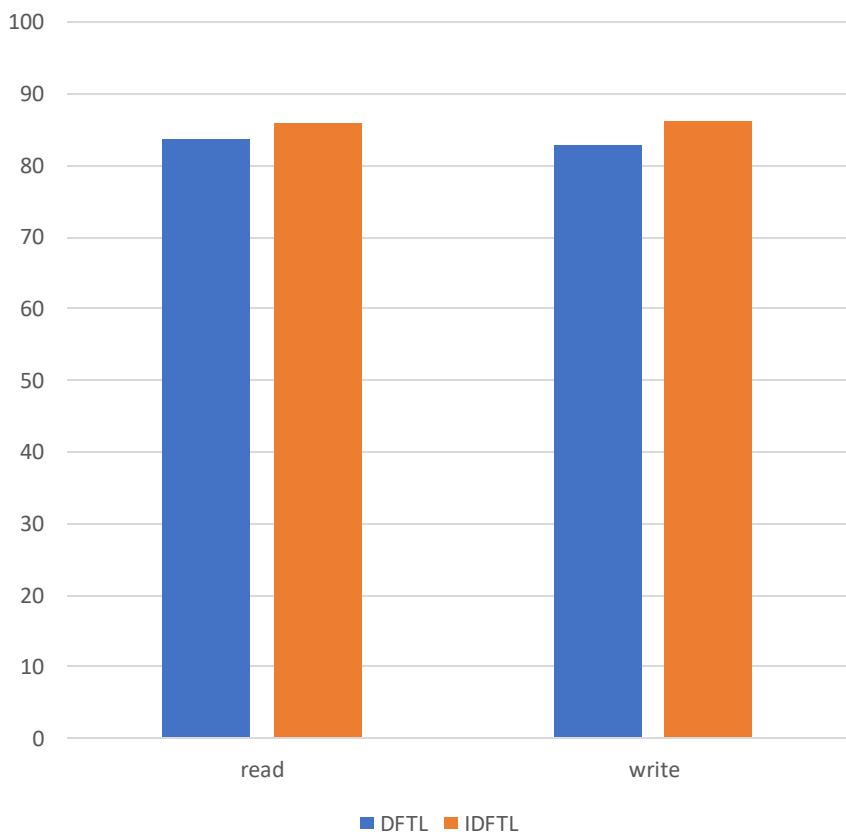
Evaluation

FIO (random read / write)

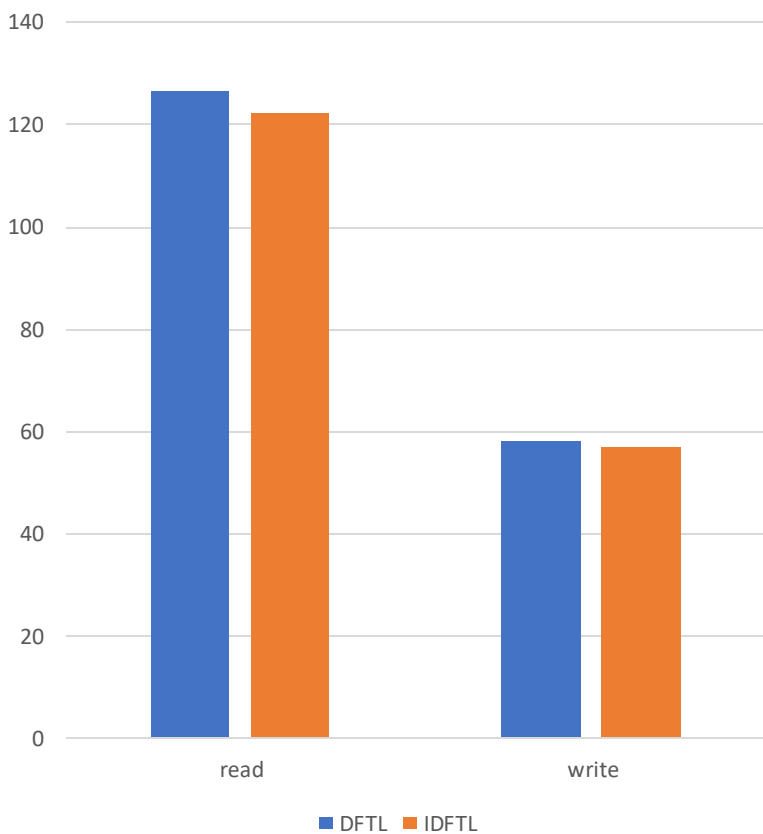
IOPS 비교 (단위: k)



Bandwidth 비교 (단위: MiB/s)



Latency 비교 (단위: μ s)



Plan

1. 코드 분석 및 구현
2. 기존 기법과 성능 비교
3. GC 관련 논문
 - The Tracing Garbage Collection Algorithm (Richard Jones and Rafael D.Lins)
 - Uniprocessor Garbage Collection Techniques (Paul R.Wilson)
4. 새로운 기법 디자인



Thank you!

Q & A ?

2024.08.28

Presentation by Dayeon Wee

wida10@dankook.ac.kr