

LxFTL

2024 FTL Study

2024. 8. 23

Presentation by Nakyeong Kim, Yejin Oh

{nkkim, yeojinoh}@dankook.ac.kr

Contents

1. Motivation
2. Plan
3. Environment
4. Conclusion

Motivation

Motivation



- LeaFTL은 한 세그먼트에서 수용할 수 있는 LPA의 개수(L)가 256으로 제한되어 있다.
- 그래프의 수렴성보다, 1~8 길이를 가지는 세그먼트가 80% 가까이 차지한다는 점에 집중했다.

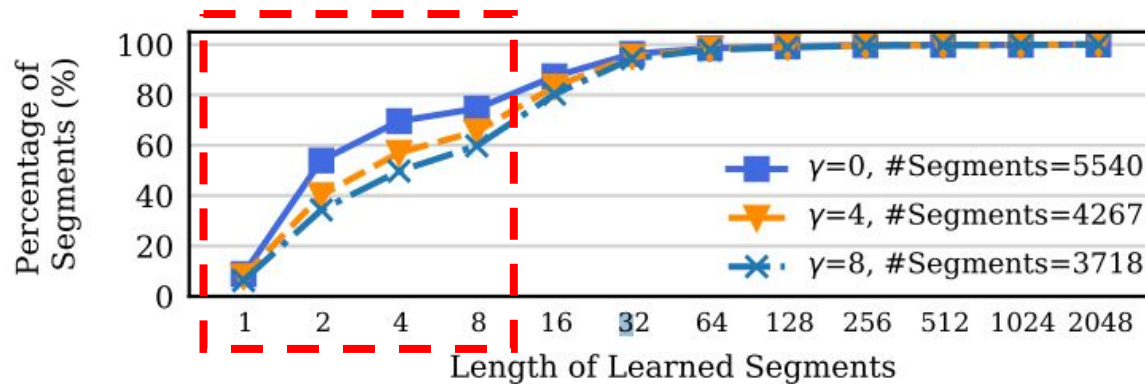
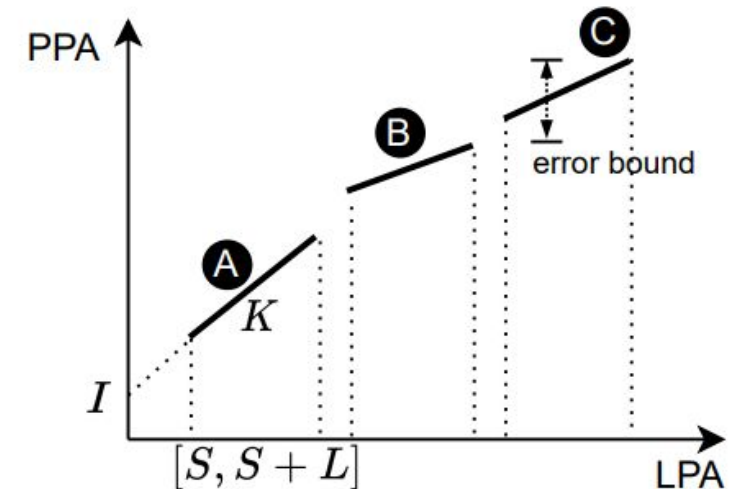


Figure 5: Aggregated distribution of learned segments.



Motivation

1. 작은 세그먼트들을 합치고
2. 길이가 256이상인 큰 sequential 데이터들을 하나의 세그먼트로 관리할 수 있다면 (길이를 동적으로 조정 가능하면)



↓
memory footprint를 더 줄일 수 있을 것

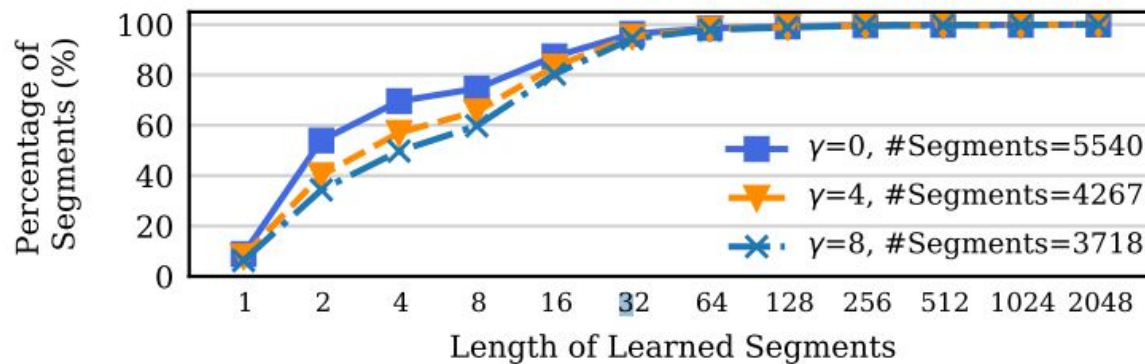
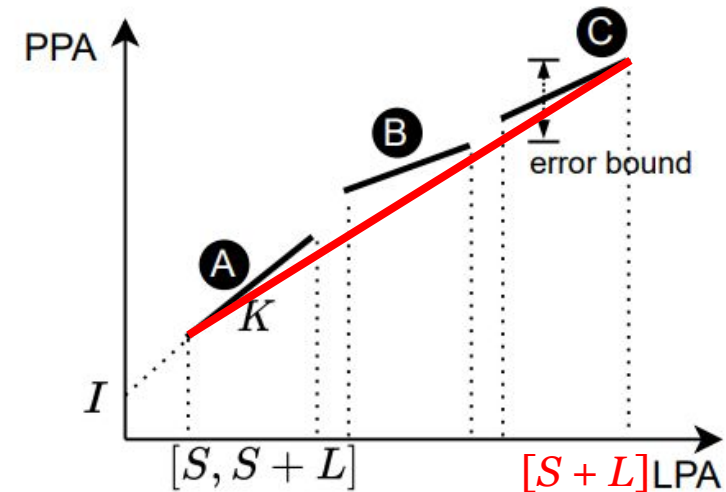


Figure 5: Aggregated distribution of learned segments.



Plan

Step

1. 코드 분석
2. 성능 평가
3. $L = 256$ 제약 조건 완화 후 성능 측정 및 이전과의 비교

1. 코드 분석
2. 성능 평가
3. $L = 256$ 제약 조건 완화 후 성능 측정 및 이전과의 비교

LeaFTL vs LearnedFTL?

1. 코드 분석
2. 성능 평가
3. $L = 256$ 제약 조건 완화 후 성능 측정 및 이전과의 비교

LeaFTL vs LearnedFTL?

Wiscsim 기반	FEMU 기반
벤치마크 O	벤치마크 X
확장성 X	확장성 O

Problem 1

LeaFTL 구현

```
class Segment():
    FPR = 0.01
    PAGE_PER_BLOCK = 256
    BITMAP = True

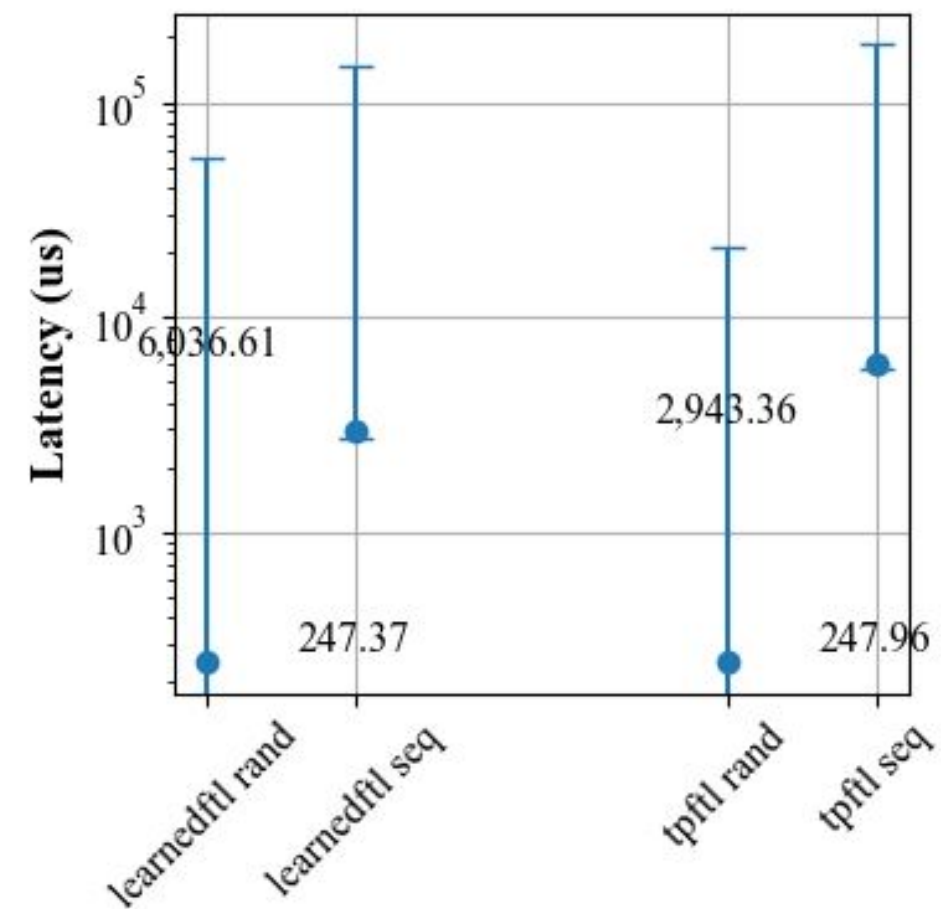
    def __init__(self, k, b, x1, x2, points=None):
        self.b = b      1
        self.k = k      1
        self.x1 = x1    2
        self.x2 = x2    4
        self.accurate = True
        self.filter = None
```

LearnedFTL 구현

```
typedef struct simple_segment {
    uint64_t slpa; // start LPA
    int len;      4
    uint8_t is_accurate;
    float k;      4
    uint64_t inter;  8
    struct simple_segment *next;
} simple_segment;
```

코드 및 실험 신뢰도 하락
직접 측정 시도

Problem 1



Step

- 1. 코드 분석
- 2. 성능 평가
- 3. L = 256 제약 조건 완화 후 성능 측정 및 이전과의 비교

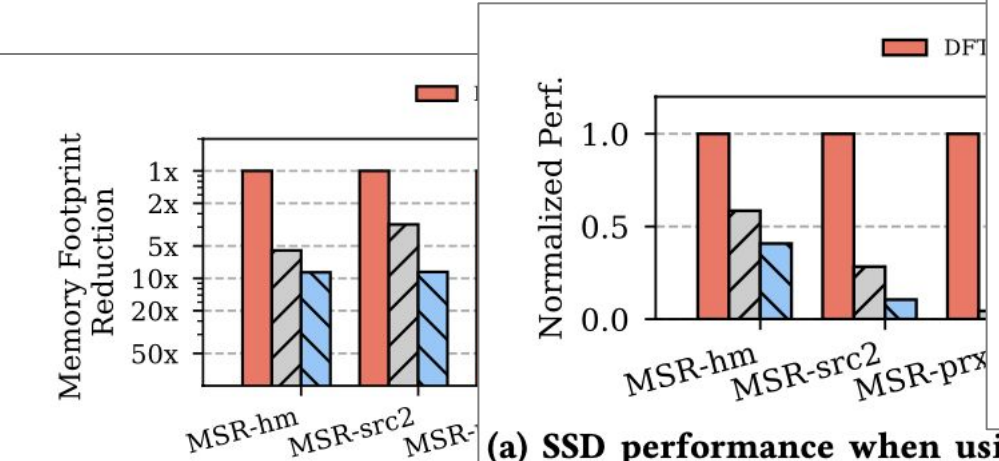
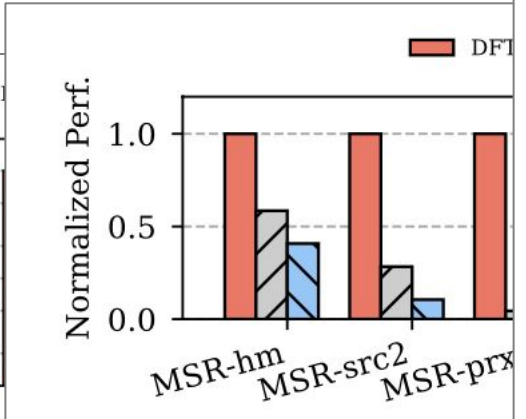


Figure 15: The reduction mapping table (lower is better).
LeaFTL, in comparison with DFTL and SFTL.



(a) SSD performance when using its DRAM mainly for the address

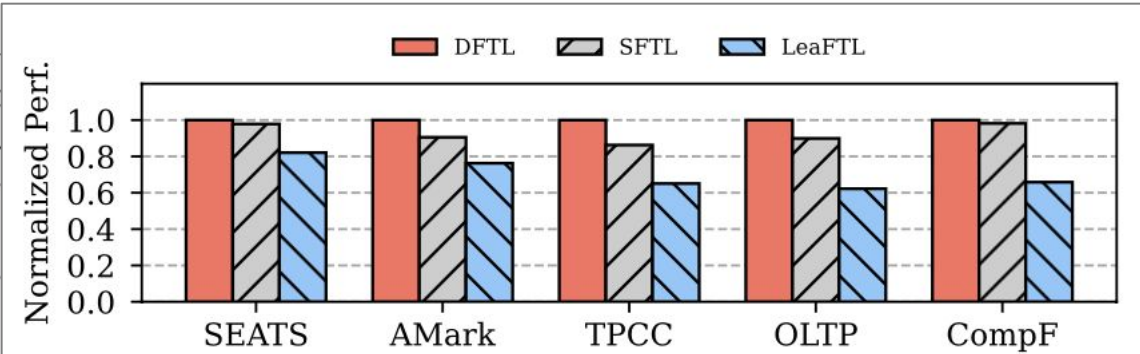
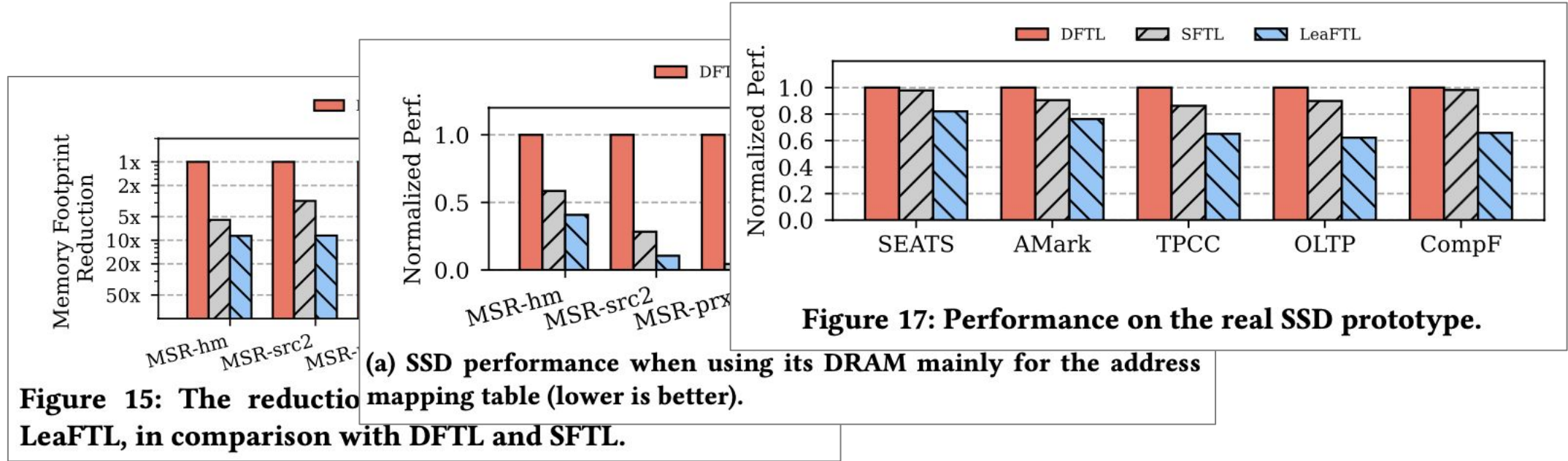


Figure 17: Performance on the real SSD prototype.

Step

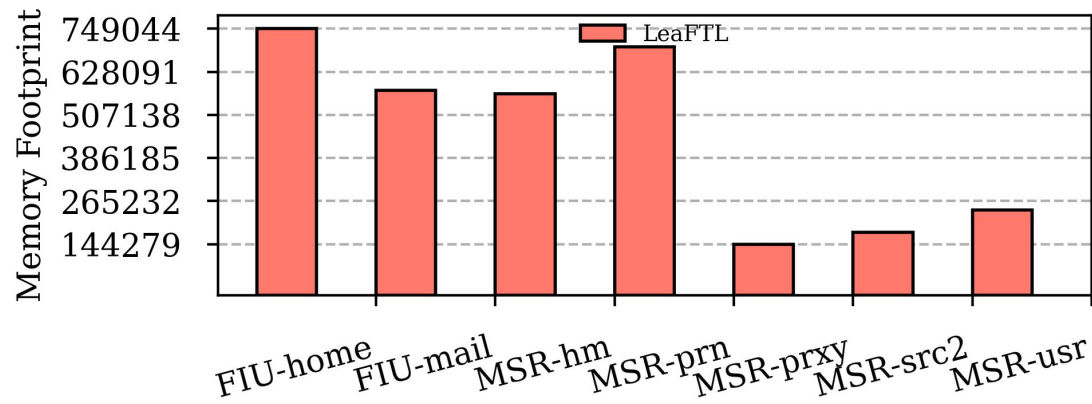
- 1. 코드 분석
- 2. 성능 평가
- 3. L = 256 제약 조건 완화 후 성능 측정 및 이전과의 비교



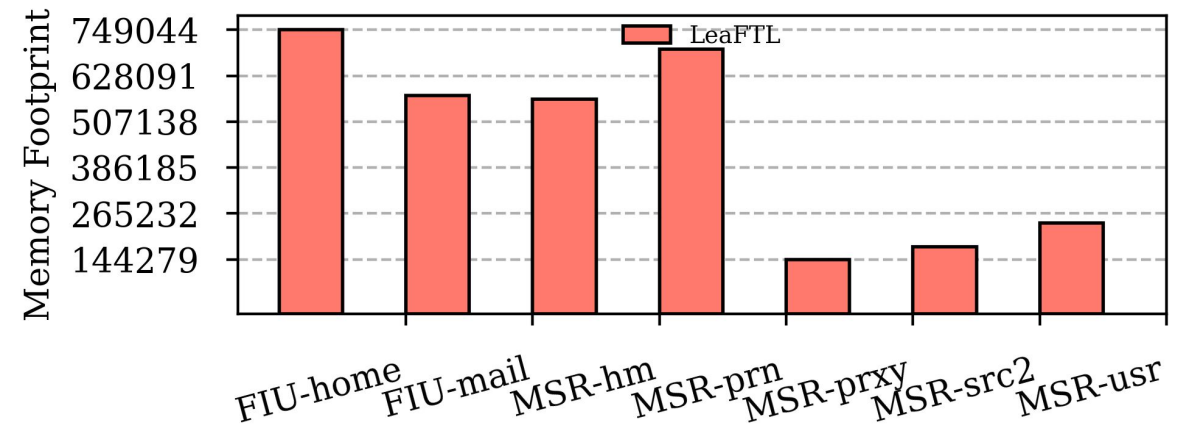
실제 값은 뭔데?
- 첨부 예정

Try LeaFTL

- wiscsim 실험 환경 구축 성공
- segment length = 256, 512 일때 비교
- 메모리 사용량에서는 크게 변하지 않음 → Approximate/accurate segment 개수 확인 필요



segment length (L) = 256



segment length (L) = 512

Try LeaFTL

LeaFTL에서 필요했던 dependency...

- ubuntu 20.04 이하 버전
- pypy 2.7
- python 3.6 이상 버전
- 20개 이상의 pip package...
 - python2.7에 맞는 낮은 버전의 패키지 설치 필요..

Try LeaFTL

다음주..

1. write amplification
2. distribution of segments (accurate/approx.)

Thank you!

Q & A ?

2022.12.10

Presentation by Choi, Gunhee

choi_gunhee@dankook.ac.kr

Appendix: Environment

Setting - LearnedFTL

Setting	Paper	Our
CPU	Intel Xeon(R) Gold 5318Y 2.10GHz	Intel(R) Xeon(R) Silver 4210R 2.40GHz
Memory	128GB	128GB
OS (Kernel)	Linux 5.4.0	Linux 5.15.0
Storage	32GB (+2GB OP)	32GB

Parameter	Value	Parameter	Value
#Channels	8	Read latency	40\mu s
#Ways	8	Write latency	200\mu s
#Blocks	256	Erase latency	2ms
#Pages	512	GTD entry group	256
Page size	4KB		

FIO Benchmark

Parameter	Value	Parameter	Value
ioengine	psync	bs	4k
iodepth	64	numjobs	64
size	410M (512 * 0.8)	rw	write
direct	1	directory	/mnt/ext4