Block Cache 크기에 따른 압축률–읽기 성능 상관관계 분석

[Storage Optimization]

딥스토리 (DBStory) 팀

소프트웨어학과 이기윤
국제경영학과 홍사인
소프트웨어학과 노승아
소프트웨어학과 성진욱

단국대학교 DANKOOK UNIVERSITY

목 차

이전 연구: 적응형 압축

적응형 압축

```
CPUS=("2" "4" "6")
MEMS=("2g" "4g" "8g")

STRATEGY_CONFIGS=(
    "adaptive:none none lz4 lz4 zstd zstd zstd"
    "all_zstd:zstd zstd zstd zstd zstd zstd zstd"
    "all_lz4:lz4 lz4 lz4 lz4 lz4 lz4 lz4"
    "no_comp:none none none none none none none"
)

export BENCHMARKS="fillseq,compact,readrandom,stats"
export NUM_KEYS=2000000
export VAL_SIZE=4096
export BLOCK_SIZE=4096
export CACHE_SIZE=$((128 * 1024 * 1024))

ITERATIONS=1
```

배경

CPU의 빠른 발전으로 압축의 비용이 감소함
더 강한 압축을 사용해도 전체 성능에 문제가 없을 수 있지 않을까 하는 생각에서 출발

가설

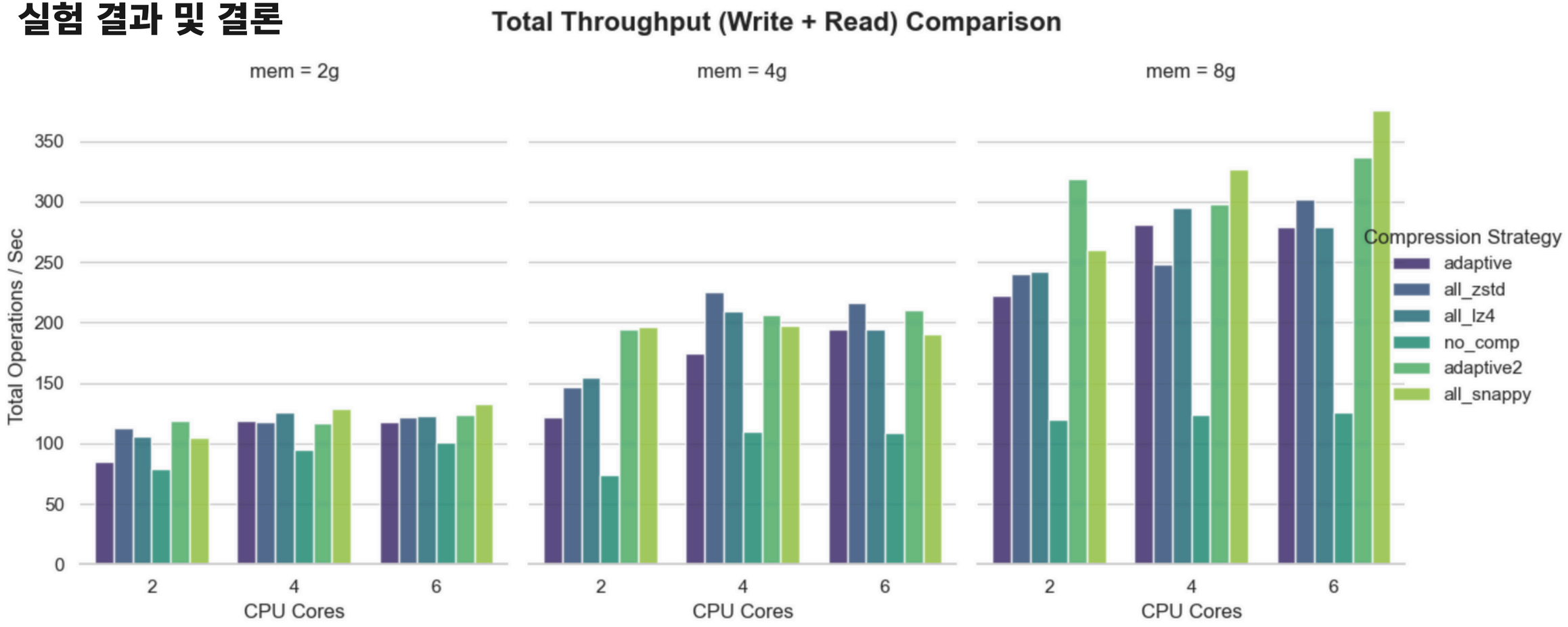
CPU의 성능이 향상됨에 따라 Compression을 강하게 하는 것이 전체적인 성능을 향상시킬 것이다.

실험 방법

코어 개수, 메모리 크기, 압축 방식 등 파라미터를 설정하여 db_bench 실행

적응형 압축

실험 결과 및 결론

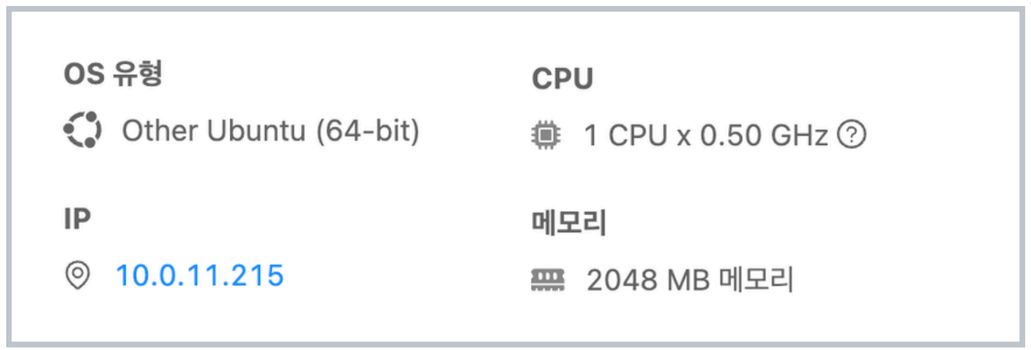


➡ CPU의 성능이 향상됨에 따라 Compression을 강하게 하는 것이 전체적인 성능을 향상시킨다.

쏠림 현상

```
echo "start_Adaptive test"

./db_bench --benchmarks="fillseq,readrandom" \
--options_file=./mixed_compression.ini \
--num=1000000 \
--value_size 1024 \
--statistics=1 \
--read_random_exp_range=200000 \
--cache_size=268435456 \
| grep -E 'readrandom|P50|P99'
```



배경

디스크 최하단의 오래된 데이터라도, 트래픽이 쏠리는 인기도 기반 쏠림 존재
→ 반복적인 압축 해제 비용 발생

가설

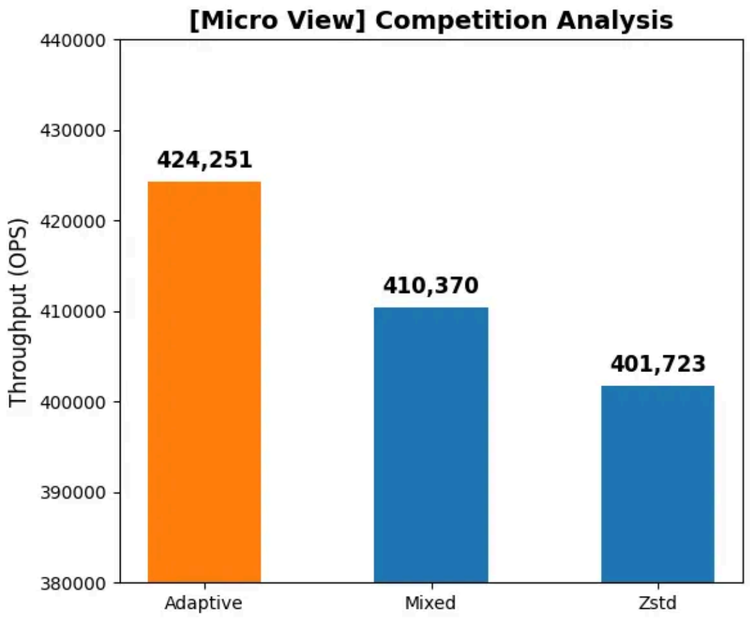
Adaptive 전략이 쏠림 현상에서도 성능 저하 없이 버틸 수 있을 것인가?

실험 방법

압축 방식과 read_random_exp_range 등의 파라미터를 설정

쏠림 현상

실험 결과 및 결론

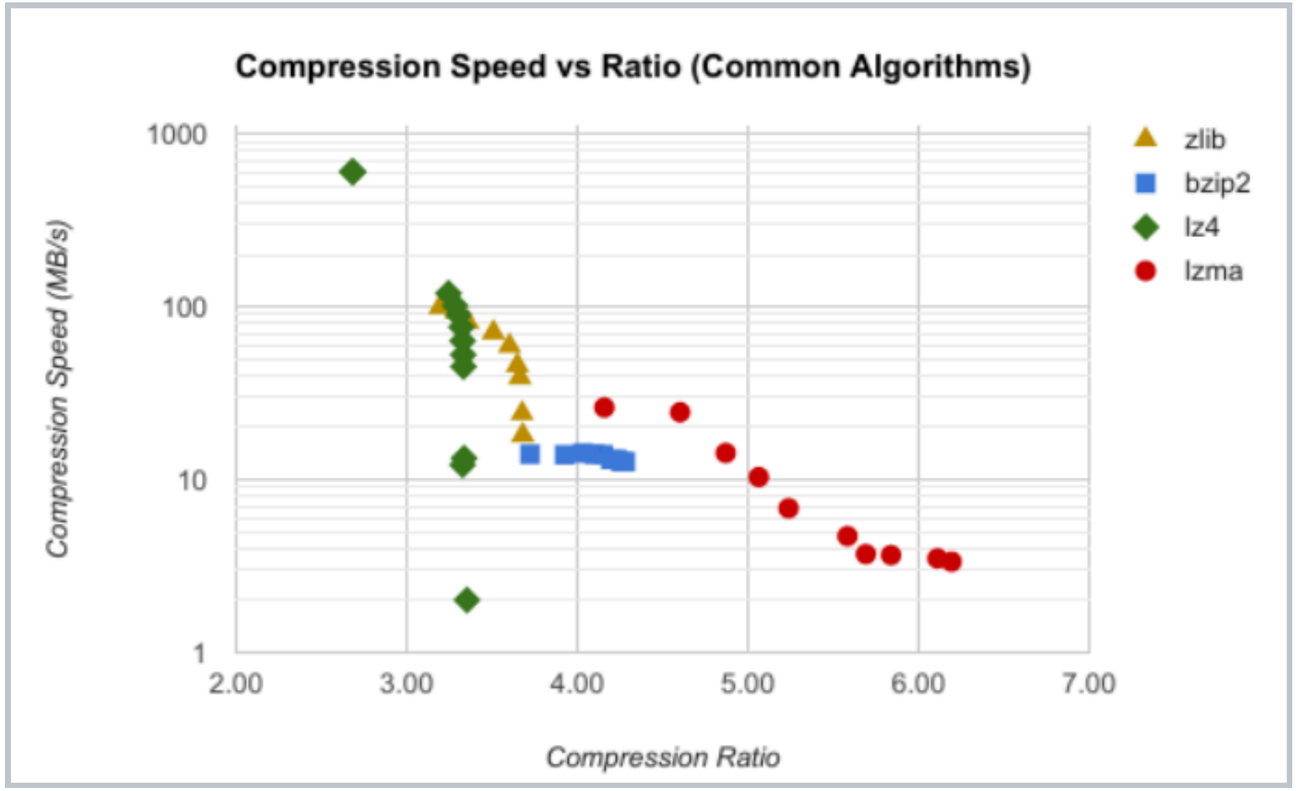


압축 전략	처리량 (OPS)	지연 시간 (Latency)	성능 비율 (vs None)
Adaptive (L0~1:None, L2~:Zstd)	424,251	2.36 µs	85%
Mixed (LZ4 + Zstd)	410,370	2.44 µs	82%
All Zstd (전체 Zstd)	401,723	2.49 µs	80%

➡ 읽기 성능은 Block Cache를 활용해 높은 속도를 낸다
캐시가 받쳐주는 한, hot 데이터에 대해 마음껏 압축을 해도 성능 손실이 없다

Block Cache – 압축률 실험

연구 배경



출처: https://gregoryszorc.com/blog/2017/03/07/better-compression-with-zstandard/

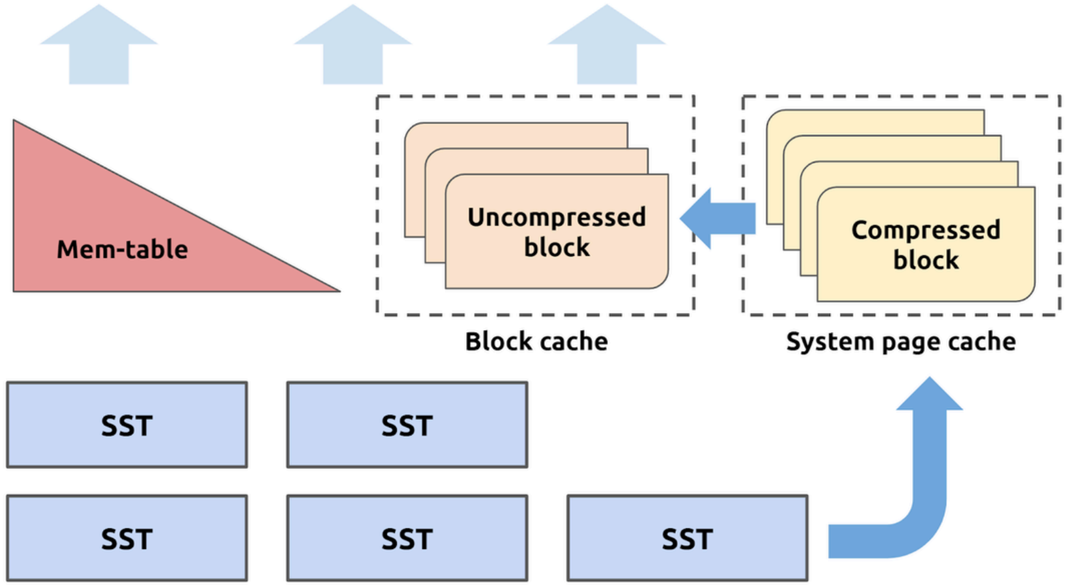
압축에 관해 더 깊게 파볼 수 있는 게 없을지 고민

압축률을 높이면 저장공간은 줄어들지만
CPU 압축/해제 비용이 증가

따라서 일반적으로
압축률 ↑ = 성능 ↓ 경향

그러나 RocksDB에는압축 해제된 데이터를
저장하는 **Block Cache**가 존재

연구 배경



출처: https://pingcap.co.jp/blog/best-practices-for-tidb-on-aws-cloud/

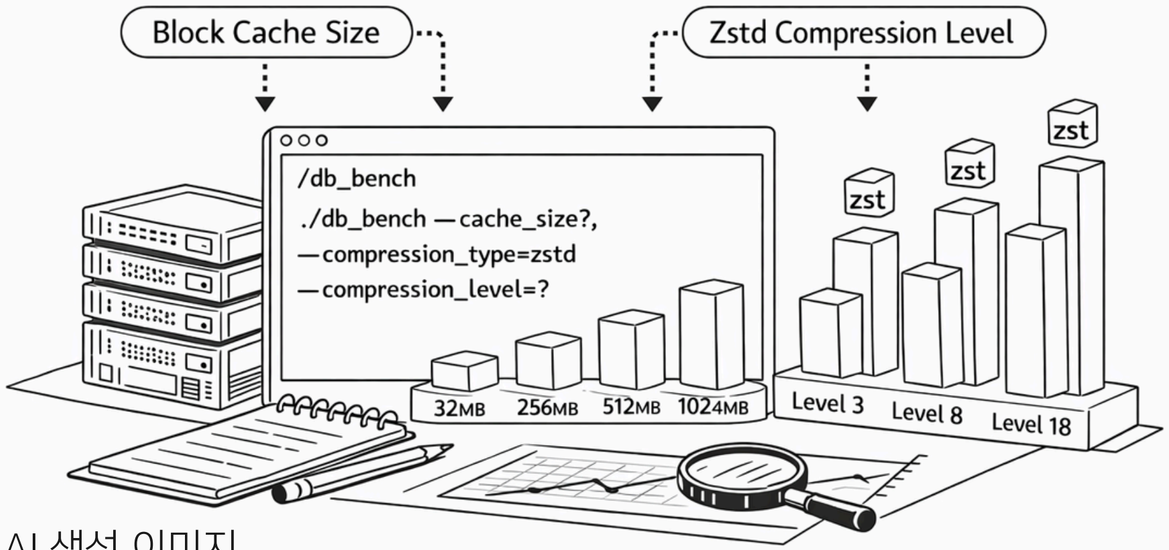
Block Cache

RocksDB에서 따로 지정할 수 있는
캐시 메모리

압축되지 않은 데이터를 배치해서 먼저 읽도록 하기 때문에 적절히 조절하면 성능을 향상시킬 수 있다

➡ 캐시가 충분하다면, 높은 압축률로 인한 읽기 성능 저하를 상쇄할 수 있을까?

실험 목적 및 방법



AI 생성 이미지

실험 목적

Block Cache 가 높아질 때
압축률에 따른 성능을 알아보기 위함

실험 방법

압축률 조절을 위해 Zstd 압축 방식 사용
→ 압축 레벨 조절 가능

실험 환경

```
CACHE_SIZES_MB=(32 64 128 256 512 1024 2048 3072)
COMPRESSION_LEVELS=(3 8 13 18)

num = 500000
value_size = 4096 (4KB)

--benchmarks=fillrandom,readrandom
```

— Apple M2, 램 8GB 로컬에서 진행

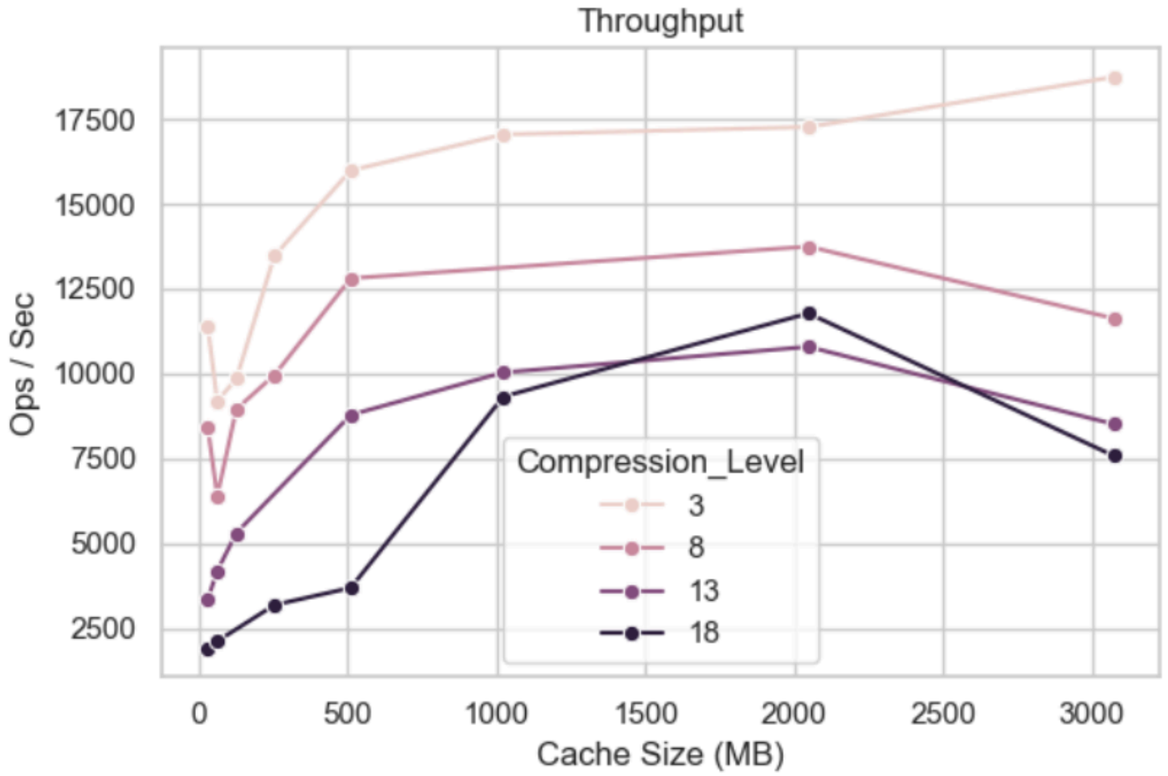
```
#CPU 코어수: 1

#OS 캐싱 제한
--use_direct_reads=true
--use_direct_io_for_flush_and_compaction=true
```

OS의 페이지 캐시를 거치지 않고
스토리지에 직접 접근

→ 페이지 캐시에 의한 왜곡 방지

결과

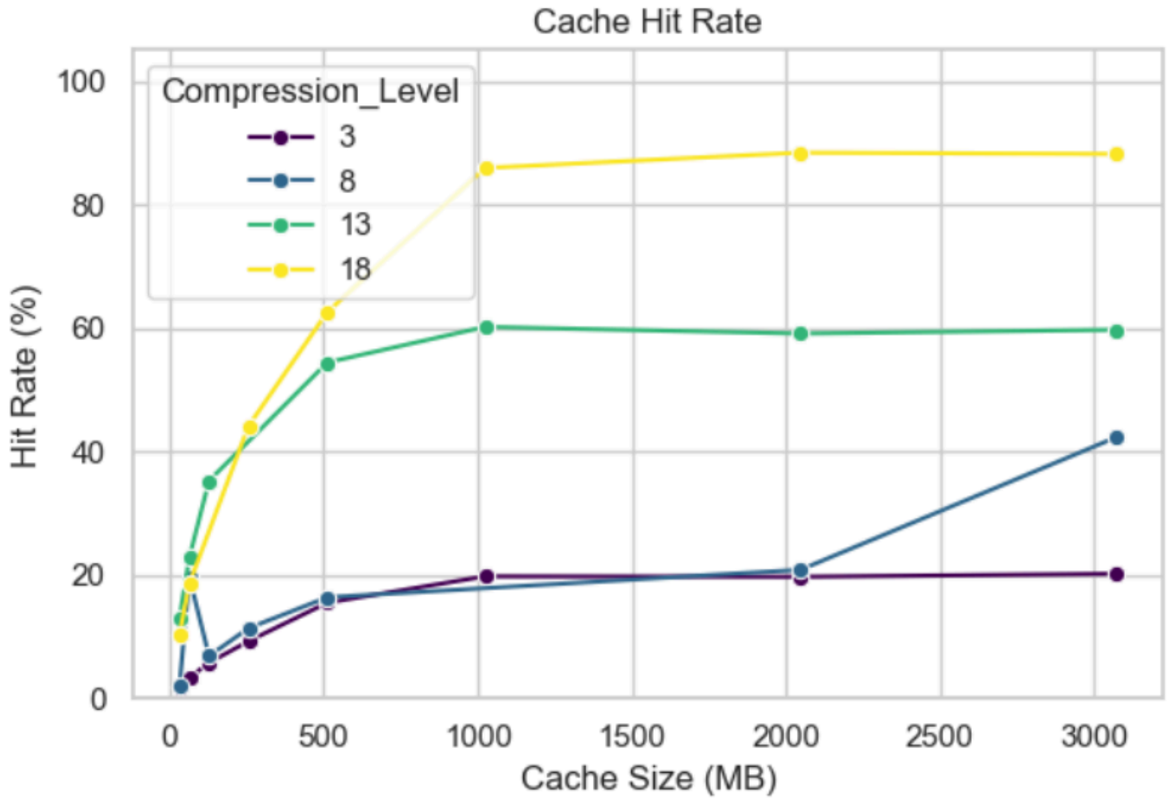


분석

캐시 사이즈와 처리량은
양의 상관관계를 보인다.

하지만 캐시 사이즈의 증가는 한계를 가진다.

결과



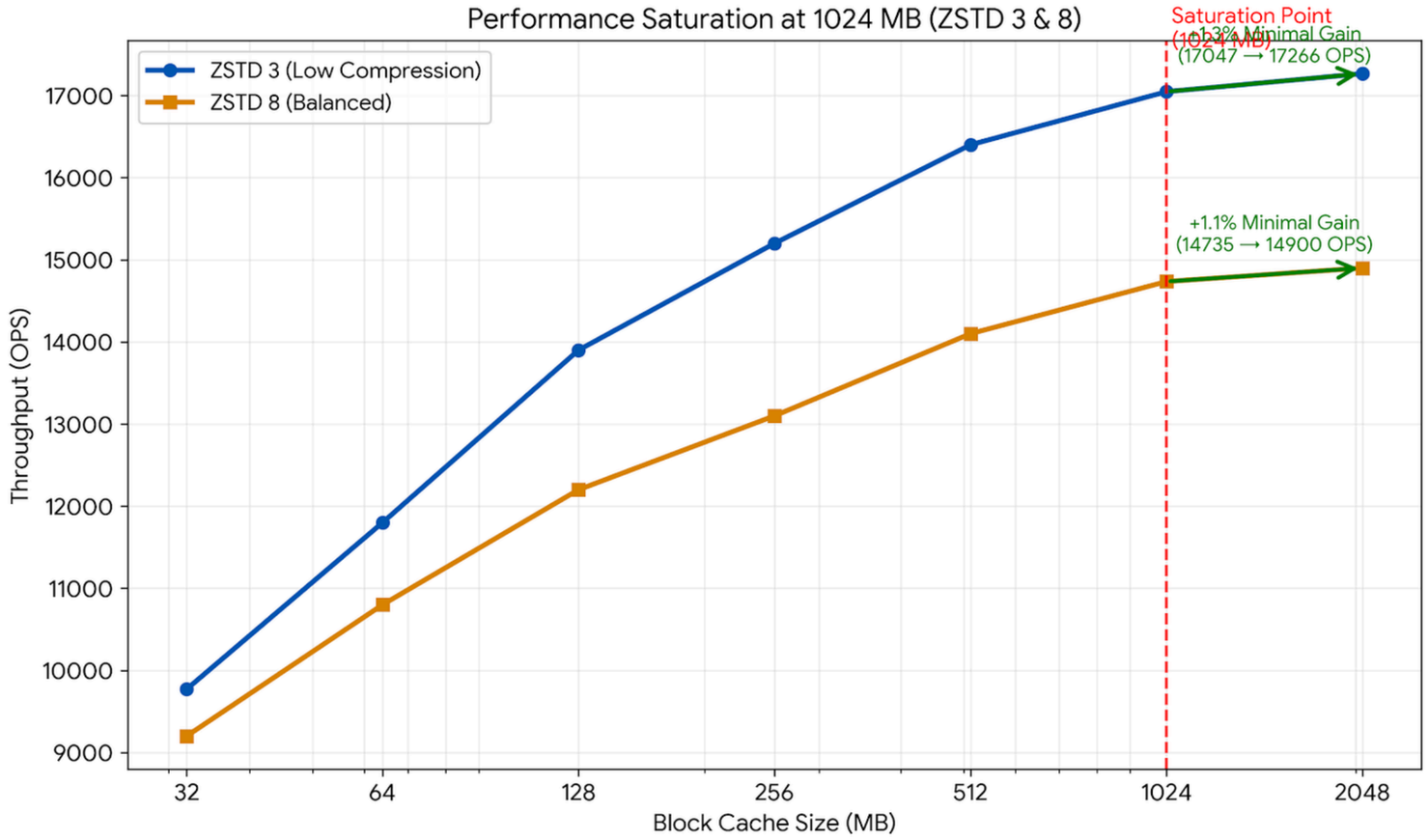
분석

캐시 사이즈와 캐시 히트는
양의 상관관계를 가진다.
하지만 이 또한 한계가 있음을 알 수 있다..

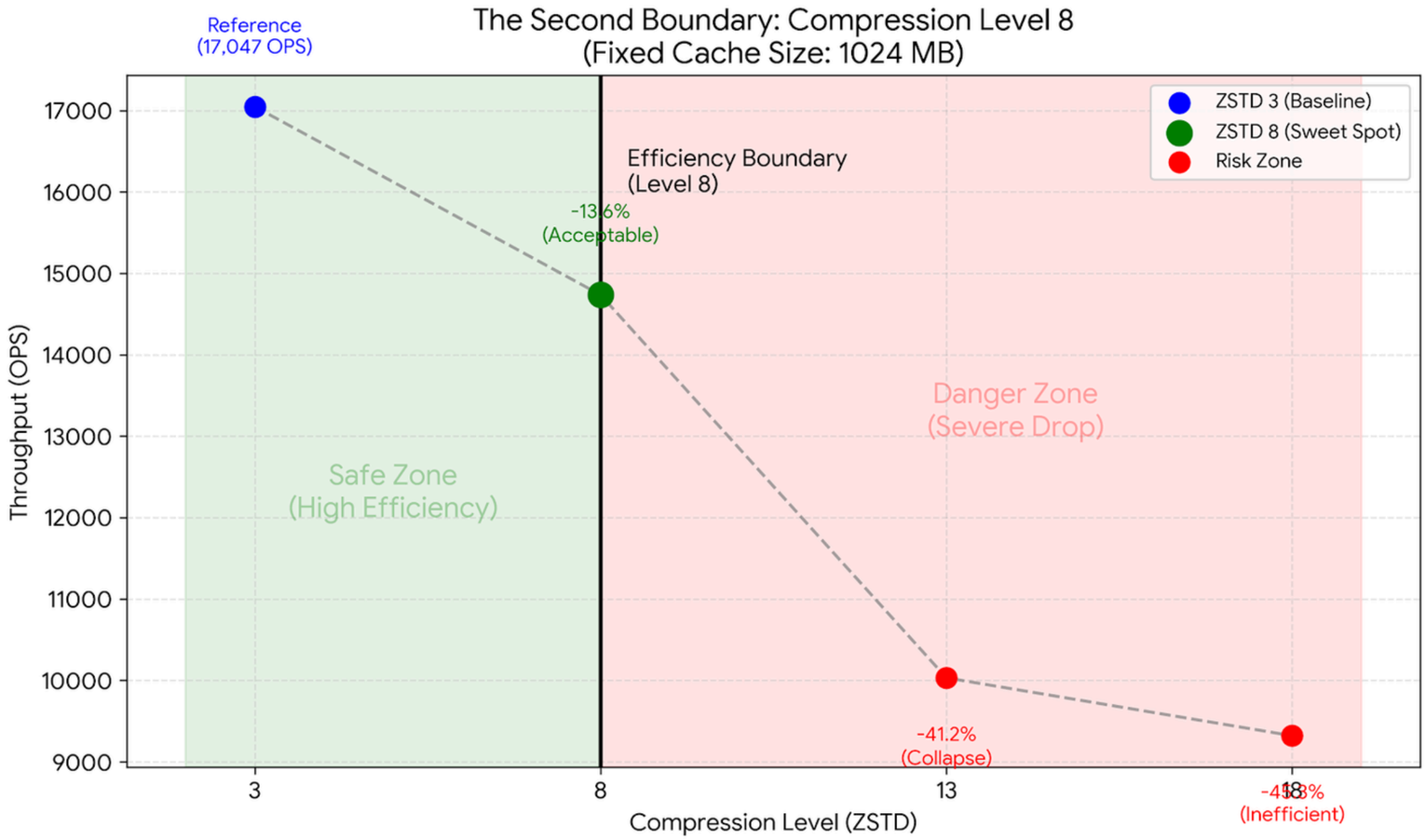
compression과 캐시 히트는
양의 상관관계를 가진다.

성능 효율의 경계(Boundary)

"Optimal Cache Size = 50% of Total Data"

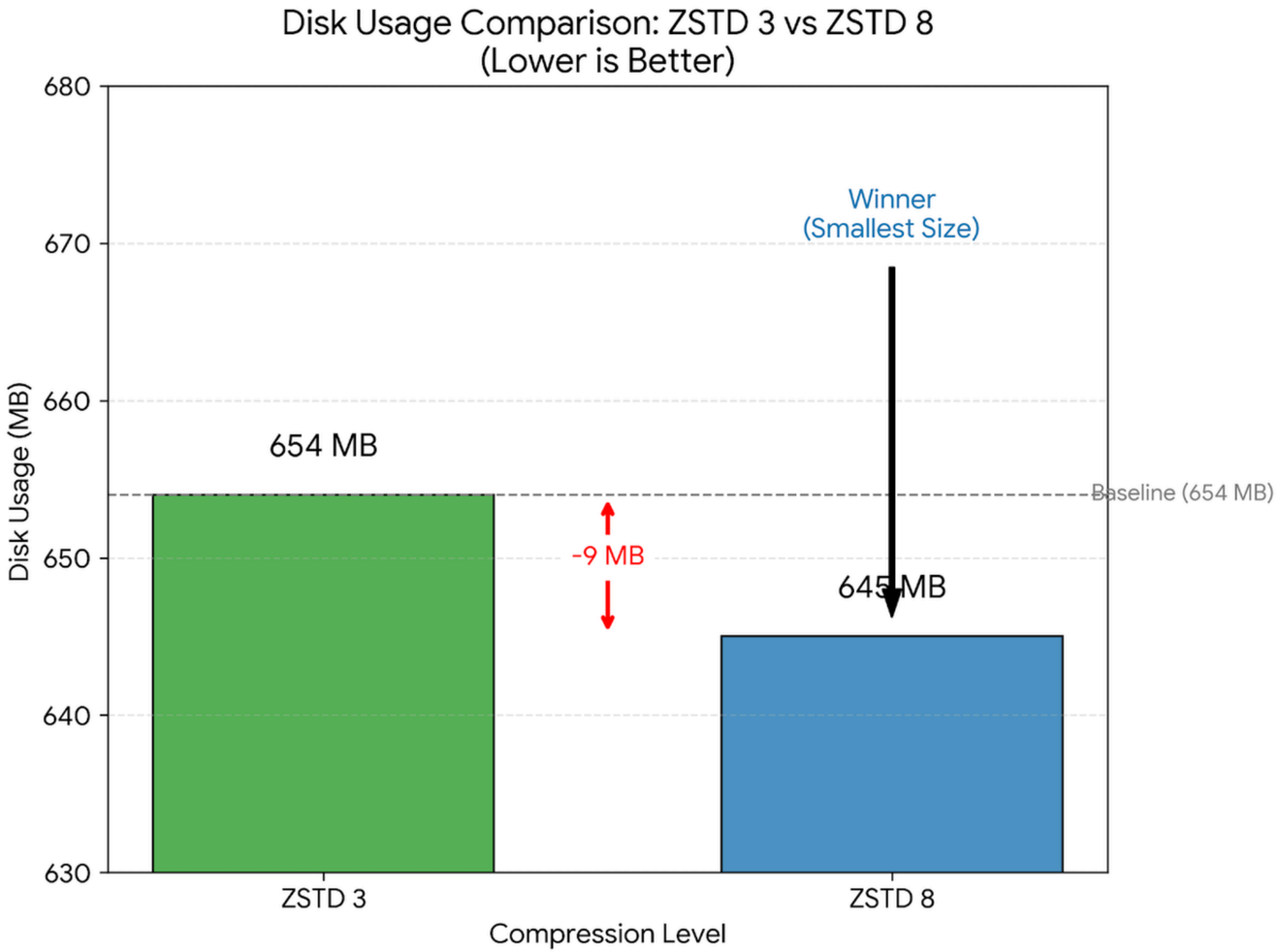


성능 효율의 경계(Boundary)



성능 효율의 경계(Boundary)

한계점 (Limitation): 미미한 용량 절감 효과 (Marginal Space Saving)



결과

Discussion

감사합니다

Block Cache 크기에 따른
압축률–읽기 성능 상관관계 분석

[Storage Optimization]

딥스토리 (DBStory) 팀

소프트웨어학과 이기윤
국제경영학과 홍사인
소프트웨어학과 노승아
소프트웨어학과 성진욱