데이터베이스 성능 향상을 위한 기계학습 기반의 RocksDB 파라미터 분석 연구

연세대학교 컴퓨터과학과 김휘군

2020년 11월

과제명: IoT 환경을 위한 고성능 플래시 메모리 스토리지 기반 인메모리 분산 DBMS 연구개발

과제번호: 2017-0-00477

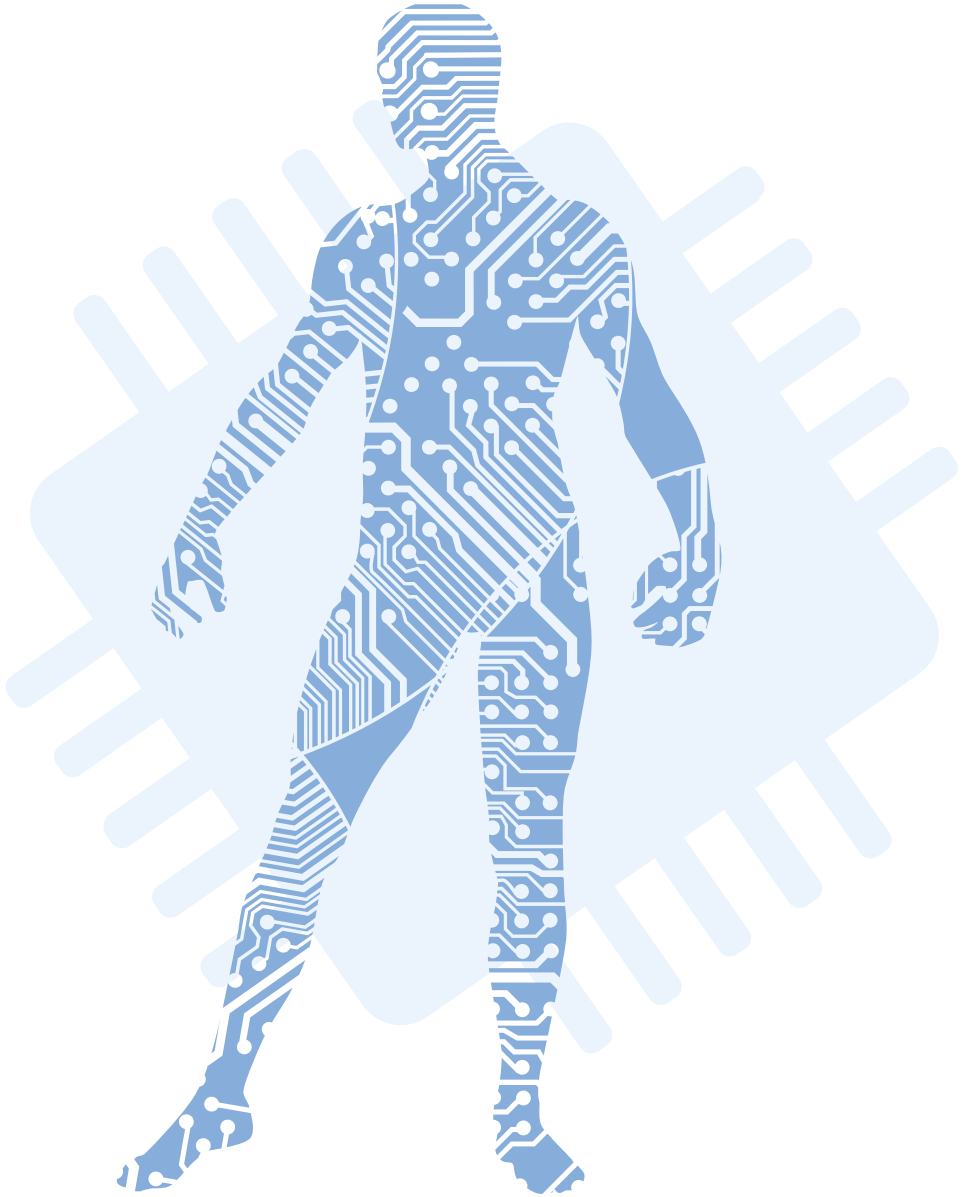
과학기술정보통신부
Ministry of Science and ICT

연세대학교
YONSEI UNIVERSITY

IITP

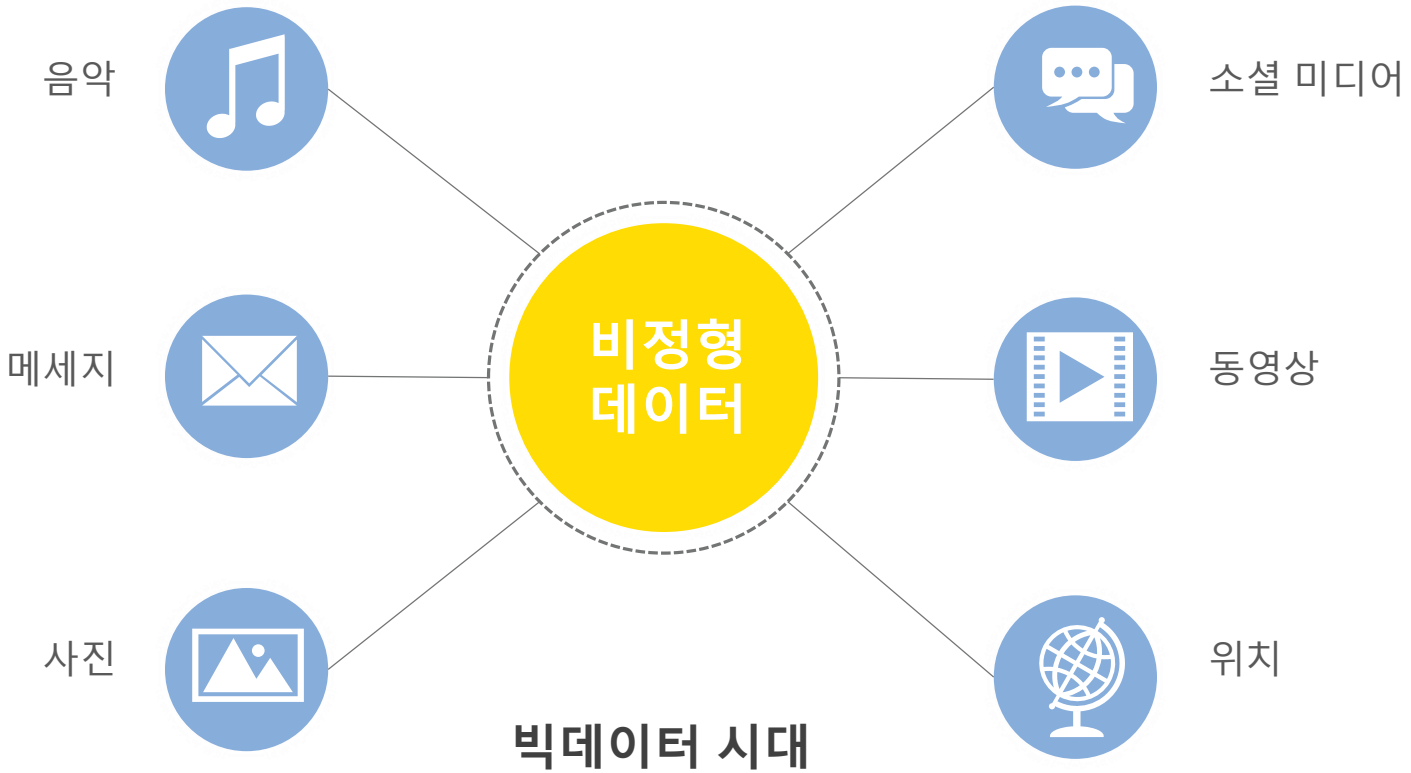
정보통신기술진흥센터
Institute for Information & communications Technology Promotion

목록



서론

서론



빅데이터 시대

키-값 쌍 형태로 저장 및 관리하는
데이터베이스 등장

서론

RocksDB

- 디스크 기반 키-값 데이터베이스
- LSM-Tree 구조 사용

LSM-Tree

- B-Tree 구조보다 우수한 쓰기 성능
- 쓰기 증폭 & 공간 증폭
    - 성능 저하
    - SSD 수명 단축
    - 데이터베이스 운영 부담 가중

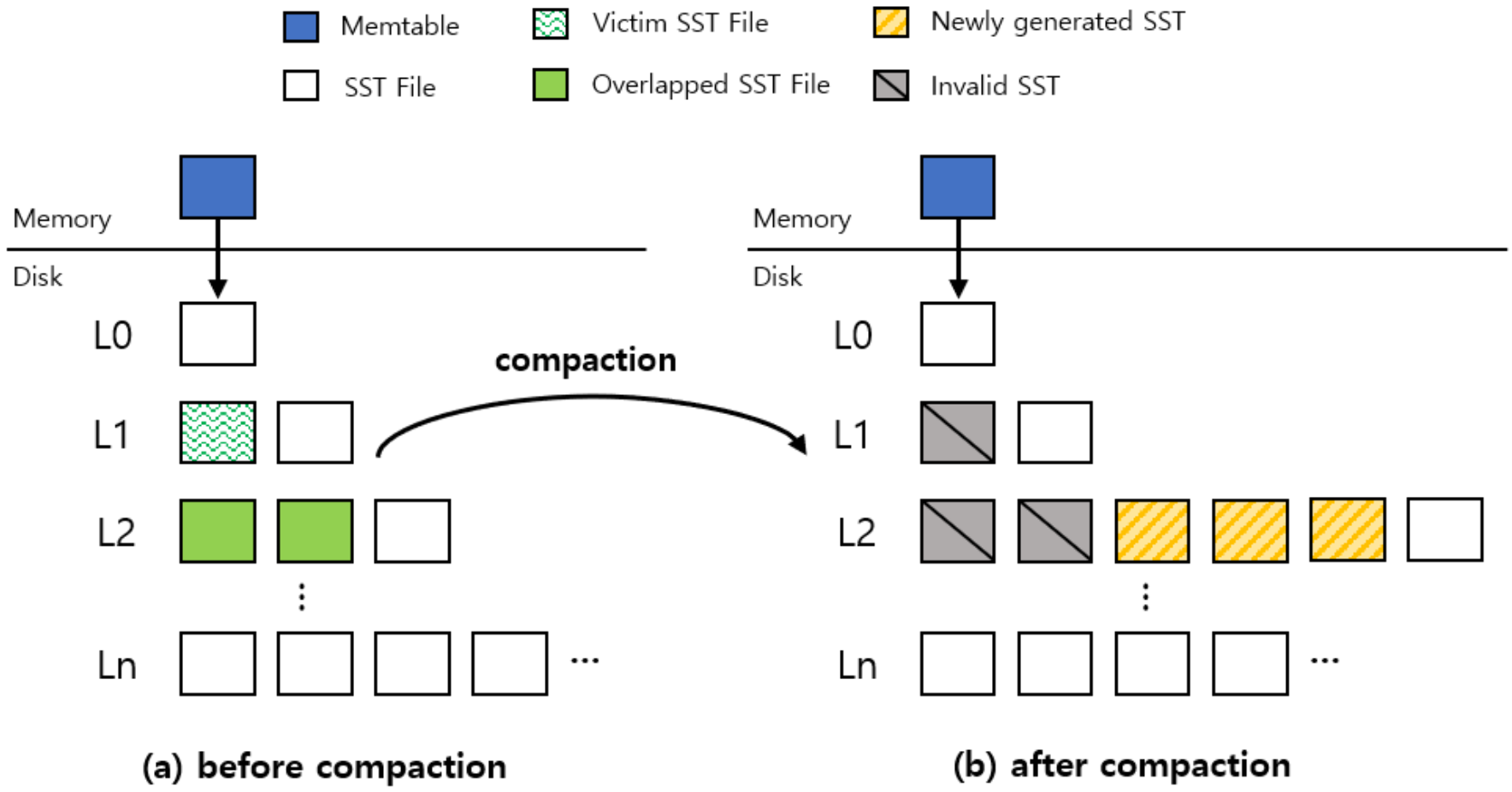
RocksDB 파라미터 최적화

- 인위적 최적화 어려움 존재
- 랜덤 포레스트
    - 중요 파라미터 식별
    - 튜닝 부하 줄임

배경

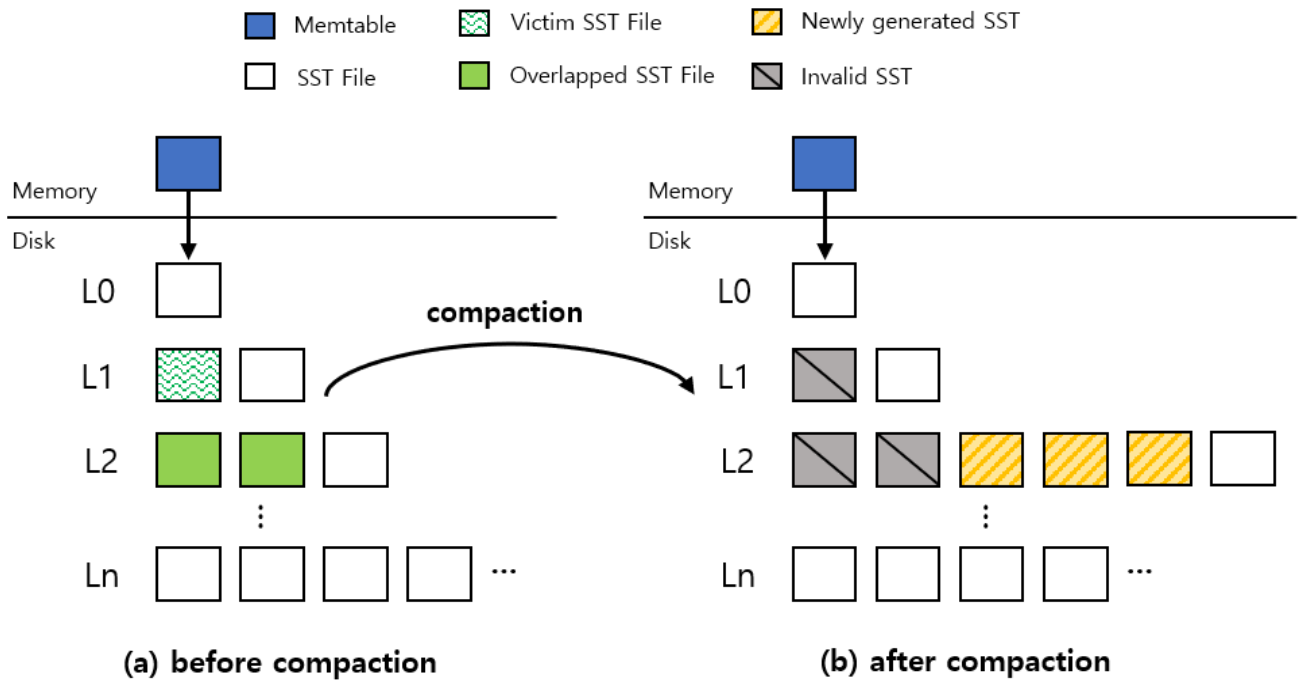
2.1. RocksDB의 데이터 저장방식
2.2. 쓰기 증폭 & 공간 증폭

2.1. RocksDB 데이터 저장방식



(그림 1) LSM-tree의 컴팩션 도식화

2.2. 쓰기 증폭 & 공간 증폭



(그림 1) LSM-tree의 컴팩션 도식화

쓰기 증폭(Write Amplification; WA)

- 데이터베이스에 저장하기 위한 추가적인 쓰기 연산
- 동시 다발적으로 발생
- Overlapped SST File

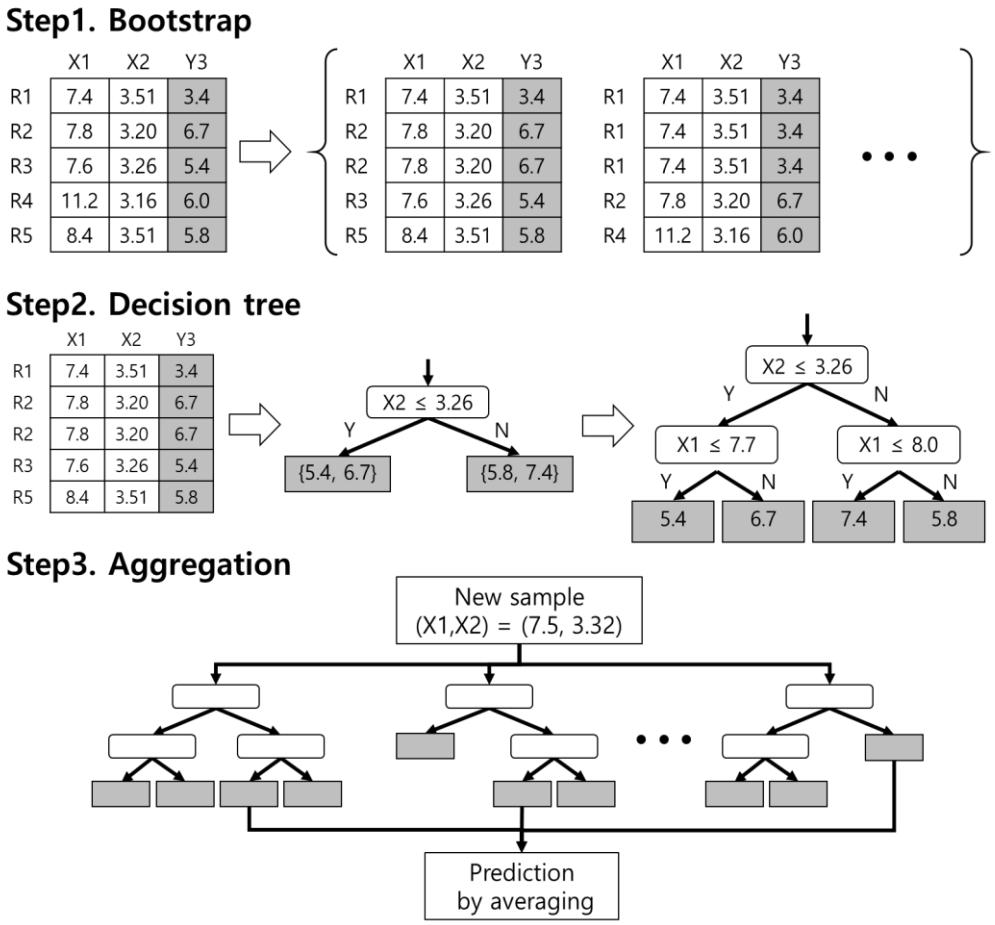
공간 증폭(Space Amplification; SA)

- 데이터베이스에 저장하기 위한 추가적인 공간 사용
- 동시 다발적으로 발생
- Invalid SST

분석 방법

3.1. 랜덤 포레스트 및 기여도 분석
3.2. 회귀모델 결정계수
3.3. 분산분석

3.1. 랜덤 포레스트 및 기여도 분석



(그림 2) 랜덤 포레스트 도식화

랜덤 포레스트 회귀모델

- 앙상블 기계학습 방법
- Step1: 부트스트랩(bootstrap)을 통해 서로 다른 훈련데이터 여러 개 생성
- Step2: 훈련데이터에 의사결정트리(decision tree) 적용
- Step3: 의사결정트리 출력 결합 및 최종출력 예측

기여도 분석

- 평균제곱오차(mean squared error)로 불순도 정의
- 불순도평균감소량(mean decrease impurity)으로 특징 기여도 산출

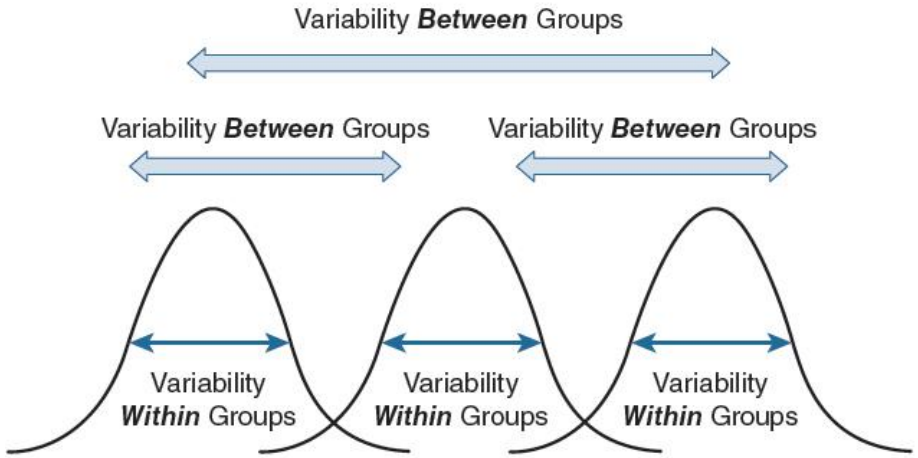
3.2. 회귀모델 결정계수

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2} \qquad (1)$$

결정계수(coefficient of determination; R^2)

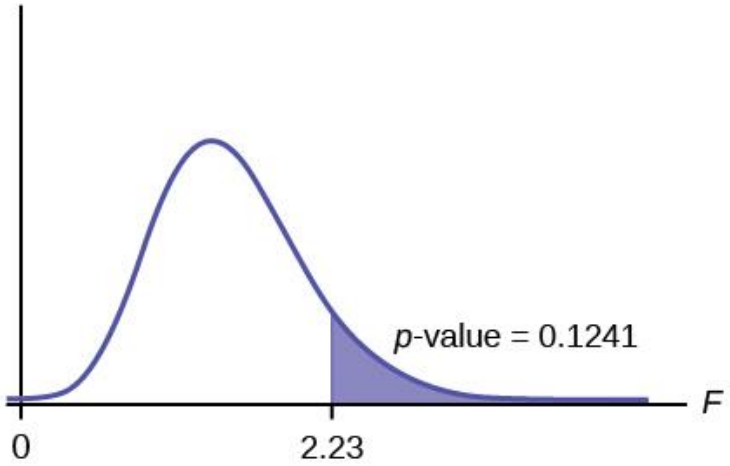
- 모델 학습 적합도 평가
- 1에 가까울수록 잘 훈련

3.3. 분산분석



(그림 3) 그룹간 및 그룹내 분산

$$F = \frac{\text{그룹간의 분산}}{\text{그룹내의 분산}} \quad (2)$$



(그림 4) p-value 검정통계그래프

분산분석(analysis of variance; ANOVA)

- 범주형 변수에 따른 성능지표 차이를 분석
- 분산의 비 F값 구함
- 검정통계치로 유의 확률(p-value) 구함
- p-value < 0.05일 때, 변수 값에 따른 성능 차이가 있다고 해석

그림 출처:
http://methods.sagepub.com/book/research-methods-statistics-for-public-nonprofit-administrators-practical-guide/i1554.xml
https://courses.lumenlearning.com/odessa-introstats1-1/chapter/facts-about-the-f-distribution/

실험환경과 실험결과 및 분석

4.1. 랜덤 포레스트 학습 평가
4.2. 파라미터 기여도 분석

실험환경

<표 1> 실험환경

OS	**CentOS 7.3.1611 (x86_64)**
CPU	Intel® Xeon® CPU E5-2660 v2 @ 2.20GHz
RAM	Samsung M393B2G70QH0-YK 16GB*4
NVMe	Intel SSDPEDMD800G4 DC P3700 800GB
RocksDB	Version 6.13

실험환경

데이터 사이즈

- 키: 16 B
- 값: 1 K
- 키-값 쌍 개수: 100만개

총: 1 G

파라미터

- compression_type
- write_buffer_size
- …

총: 30 개

반복 횟수

- 무작위 파라미터 조합

총: 1000 번

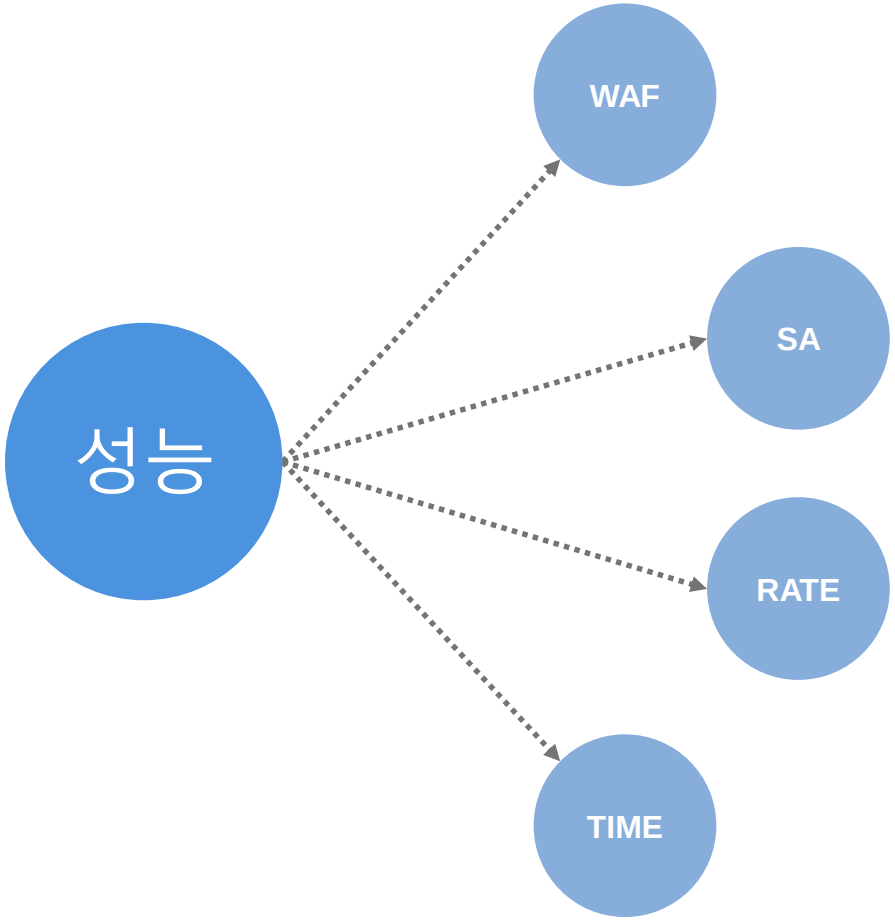
LSM-Tree 사이즈

- 데이터 생성시간 단축
- 쓰기 증폭 빈번 출현
- 공간 증폭 빈번 출현

1/64 배로 축소

실험환경

$$WAF = \frac{Bytes\ written\ to\ Storage}{Bytes\ written\ to\ Database} \quad (3)$$



WAF

Write Amplification Factor (WAF)
- 쓰기 증폭의 비율 나타냄

SA

SA
- 데이터베이스에 기록된 최종 데이터 사이즈
- 압축 알고리즘 때문에 비율 사용하지 않음

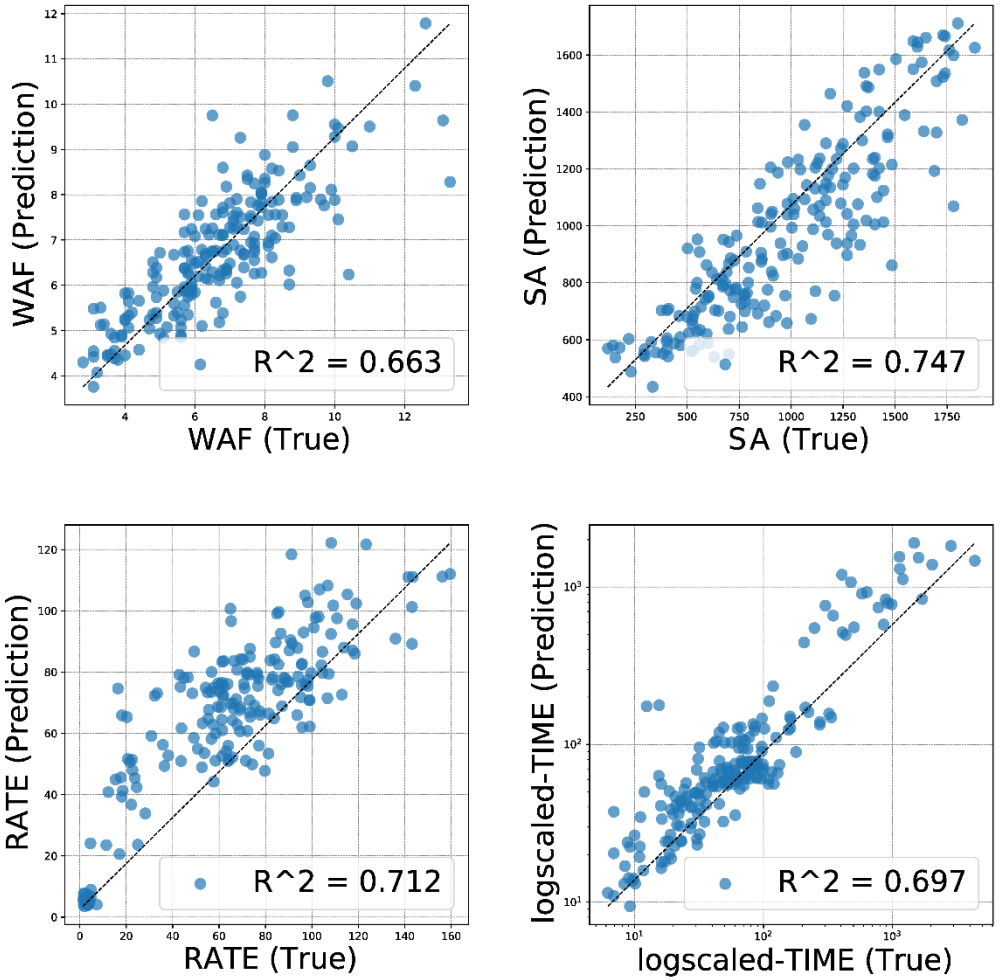
RATE

RATE
- 데이터베이스에 기록하는 평균속도

TIME

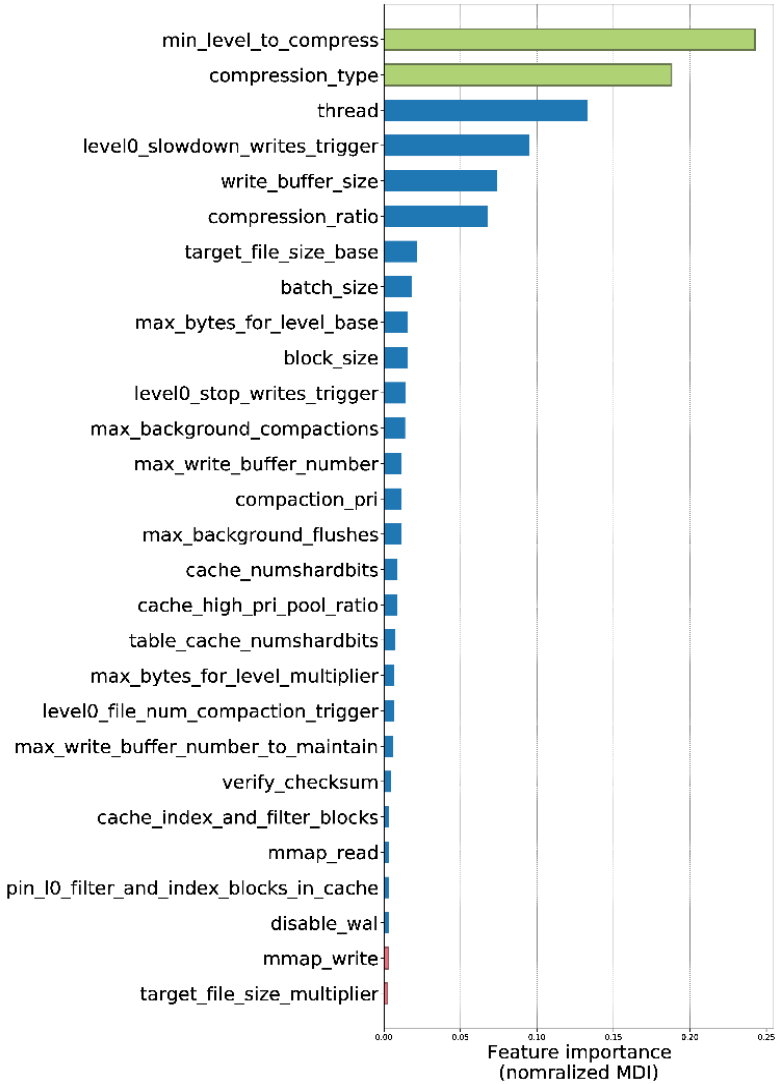
TIME
- 데이터베이스 기록의 소요시간

4.1. 랜덤 포레스트 학습 평가



(그림 5) RocksDB 성능 지표별 예측 정확도

4.2. 파라미터 기여도 분석

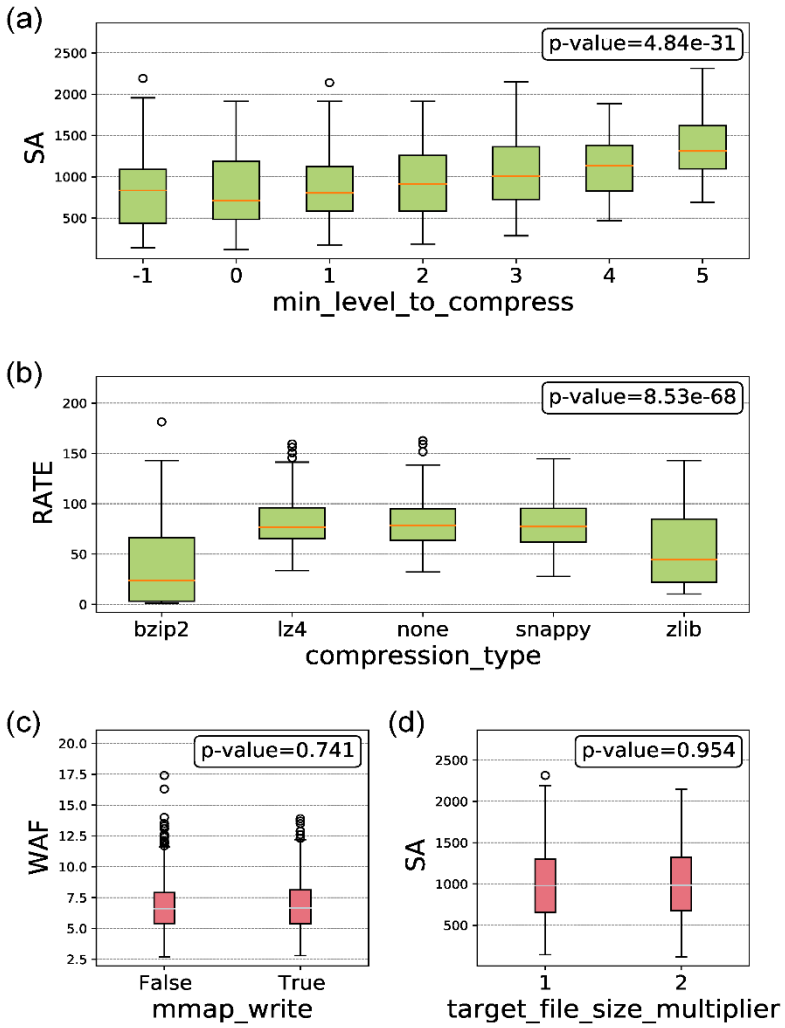


(그림 6) 지니평균감소량 기반의 파라미터 기여도

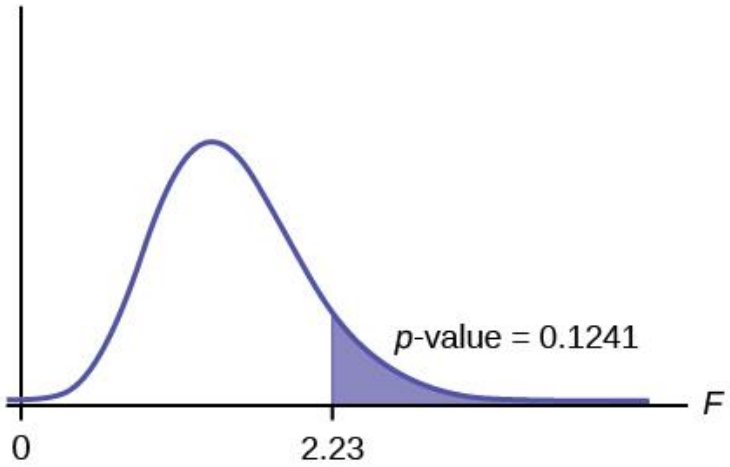
파라미터 기여도(최상위 6개)

- min_level_to_compress: 압축시작하는 계층 설정
- compression_type: 압축알고리즘 선택
- thread: 작업 수행할 스레드 개수 설정
- level0_slowdown_writes_trigger: 쓰기 속도 저하시키는 L0 파일 개수
- write_buffer_size: memtable의 사이즈
- compression_ratio: 데이터 압축률

4.2. 파라미터 기여도 분석



(그림 7) 최상위 최하위 파라미터 비교분석



(그림 4) p-value 검정통계그래프

분산분석(최상위 2개, 최하위 2개)

- min_level_to_compress: 61.7% 차이
- compression_type: 119.2% 차이
- mmap_write: 3.77% 차이
- target_file_size_multiplier: 0.51% 차이

결론

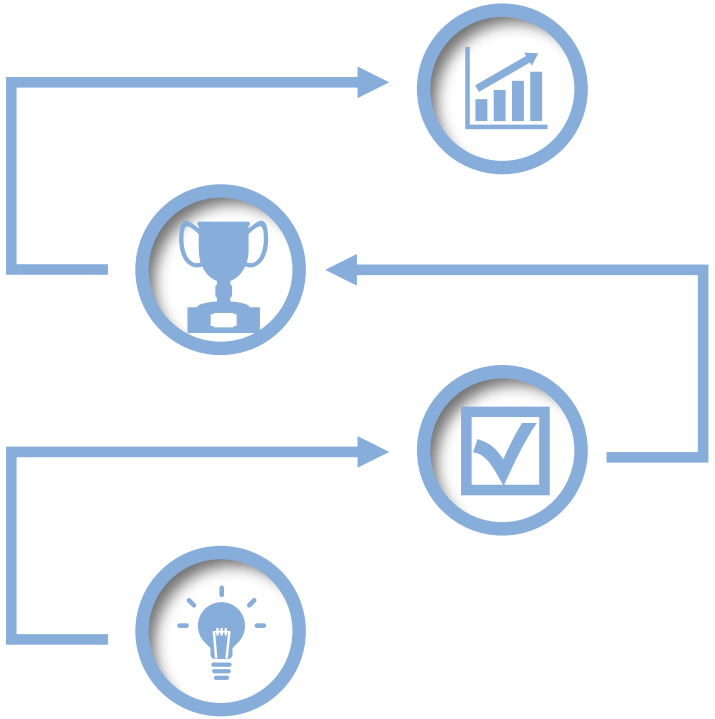
결론

향후 연구

최적화 알고리즘 기반의 RocksDB 중요 파라미터 자동 최적화 시스템 연구개발

중요한 파라미터 추출

랜덤 포레스트로 분석 및 식별하고, 분산분석으로 이 파라미터들과 성능의 연관성 확인



기여

중요 파라미터 값들을 잘 선택하면, RocksDB와 모든 LSM-Tree 사용하는 데이터베이스의 성능 제고에 도움 되고, 저장 디바이스 수명단축 완화


Thank You