

**【서지사항】****【서류명】** 특허출원서**【출원구분】** 특허출원**【출원인】****【명칭】** 연세대학교 산학협력단**【특허고객번호】** 2-2005-009509-9**【대리인】****【명칭】** 특허법인(유한)아이시스**【대리인번호】** 9-2016-100121-4**【지정된변리사】** 김형상**【포괄위임등록번호】** 2020-009519-7**【발명의 국문명칭】** 노브 벡터 표현을 이용한 데이터베이스의 최적화 방법**【발명의 영문명칭】** DATABASE TUNING METHOD USING KNOB VECTOR**【발명자】****【성명】** 박상현**【성명의 영문표기】** PARK, Sang Hyun**【주민등록번호】** 670101-1XXXXXX**【우편번호】** 08004**【주소】** 서울특별시 양천구 오목로 300, 204동 3701호(목동, 현대하  
이페리온2)**【발명자】****【성명】** 이지은

【성명의 영문표기】 LEE, Ji Eun  
 【주민등록번호】 960912-2XXXXXX  
 【우편번호】 03716  
 【주소】 서울특별시 서대문구 동교로 291, 101동 1004호(연희동, 연희동임광아파트)

## 【발명자】

【성명】 최종환  
 【성명의 영문표기】 CHOI, Jong Hwan  
 【주민등록번호】 910226-1XXXXXX  
 【우편번호】 21090  
 【주소】 인천광역시 계양구 봉오대로691번길 4, 103동 409호(작전동, 코오롱아파트)

## 【발명자】

【성명】 서상민  
 【성명의 영문표기】 SEO, Sang Min  
 【주민등록번호】 930507-1XXXXXX  
 【우편번호】 63272  
 【주소】 제주특별자치도 제주시 고마로 44, 801호(일도이동, 서진하이츠)

【출원언어】 국어

【심사청구】 청구

【공지예외적용대상증명서류의 내용】

【공개형태】 논문공개

【공개일자】 2022.09.21

【이 발명을 지원한 국가연구개발사업】

【과제고유번호】 1711103018

【과제번호】 2017-0-00477-004

【부처명】 과학기술정보통신부

【과제관리(전문)기관명】 정보통신기획평가원

【연구사업명】 SW컴퓨팅산업원천기술개발(R&D)

【연구과제명】 (SW 스타랩) IoT 환경을 위한 고성능 플래시 메모리 스토리  
지 기반 인메모리 분산 DBMS 연구개발

【기여율】 1/1

【과제수행기관명】 연세대학교 산학협력단

【연구기간】 2022.01.01 ~ 2022.12.31

【취지】 위와 같이 특허청장에게 제출합니다.

대리인 특허법인(유한)아이시스

(서명 또는 인)

【수수료】

【출원료】 0 면 46,000 원

【가산출원료】 33 면 0 원

【우선권주장료】 0 건 0 원

【심사청구료】 11 항 627,000 원

【합계】 673,000 원

【감면사유】 전담조직(50%감면)[1]

【감면후 수수료】 336,500 원

## 【발명의 설명】

### 【발명의 명칭】

노브 벡터 표현을 이용한 데이터베이스의 최적화 방법 {DATABASE TUNING METHOD USING KNOB VECTOR}

### 【기술분야】

【0001】 이하 설명하는 기술은 데이터베이스 최적화 기법이다. 특히 이하 설명하는 기술은 데이터베이스의 최적 파라미터 산출 기법이다.

### 【발명의 배경이 되는 기술】

【0002】 DBMS(database management system)는 다양한 서비스 애플리케이션에서 활용되고 있다. RocksDB는 키-값(key-value) 쌍의 형태로 비정형 데이터를 관리할 수 있는 데이터베이스이다. RocksDB는 노브(knob)의 값을 최적화하여 성능을 향상될 수 있다.

### 【선행기술문헌】

### 【특허문헌】

【0003】 (특허문헌 0001) 한국공개특허 제10-2022-0102551호

### 【발명의 내용】

### 【해결하고자 하는 과제】

【0004】 노브 값의 최적화는 대부분 사용자 경험에 따라 수행되고 있다. 한편, RocksDB는 대량의 노브들을 갖고 있어서 노브 값의 최적화가 쉽지 않다.

【0005】 이하 설명하는 기술은 자동으로 노브 값의 최적화를 수행하는 기법을 제공하고자 한다.

### 【과제의 해결 수단】

【0006】 노브 벡터 표현을 이용한 데이터베이스의 최적화 방법은 컴퓨터 장치는 데이터베이스의 타깃 워크로드의 정보를 입력받는 단계, 상기 컴퓨터 장치는 상기 타깃 워크로드에 대한 샘플들을 선별하는 단계 및 상기 컴퓨터 장치는 유전 알고리즘을 이용하여 상기 데이터베이스에 대한 최적 설정값을 결정하는 단계를 포함한다. 상기 최적 설정값을 결정하는 단계에서 상기 컴퓨터 장치가 상기 유전 알고리즘에서 사용되는 염색체에 해당하는 노브들의 희소 벡터를 산출하고, 상기 희소 벡터를 입력받아 신경망이 산출하는 내부 메트릭스를 이용하여 설정 메트릭스하고, 상기 컴퓨터 장치가 상기 설정 메트릭스를 입력받는 학습모델이 출력하는 예측값 및 상기 타깃 워크로드의 디폴트값으로 결정되는 적합도를 기준으로 상기 최적 설정값을 결정한다.

【0007】 데이터베이스의 최적 파라미터를 결정하는 컴퓨터 장치는 RocksDB의 타깃 워크로드의 정보를 입력받는 인터페이스 장치, 노브들의 희소 벡터를 입력받아 데이터베이스의 내부 메트릭스를 산출하는 제1 신경망 및 노브들의 설정 메트릭스를 입력받아 데이터베이스의 외부 메트릭스를 산출하는 제2 신경망을 저장하는

저장장치 및 유전 알고리즘을 이용하여 상기 RocksDB의 최적 설정값을 결정하는 연산장치를 포함한다. 상기 내부 메트릭스는 RocksDB의 상태를 나타내는 항목으로 구성되고, 상기 외부 메트릭스는 RocksDB의 성능을 나타내는 항목으로 구성된다.

### 【발명의 효과】

【0008】 이하 설명하는 기술은 대량의 노브들을 처리하는 학습모델을 이용하여 RocksDB와 같은 데이터베이스의 최적화를 빠르고 정확하게 수행한다.

### 【도면의 간단한 설명】

【0009】 도 1은 데이터베이스의 최적 파라미터를 예측하는 개략적인 과정의 예이다.

도 2는 데이터베이스의 노브 특징을 학습하는 과정에 대한 예이다.

도 3은 다양한 개수의 샘플 모집단을 이용하여 타깃 워크로드와 유사한 워크로드를 찾는 결과이다.

도 4는 노브 최적화 기법의 성능을 평가한 결과이다.

도 5는 예측 모델의 유형에 따른 노브 최적화 성능을 비교한 결과이다.

도 6은 노브 최적화를 추천하는 컴퓨터 장치에 대한 예이다.

### 【발명을 실시하기 위한 구체적인 내용】

【0010】 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이

아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

【0011】 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

【0012】 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설명된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

【0013】 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에

도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

【0014】 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

【0016】 이하 설명하는 기술은 데이터베이스의 노브를 최적화(tuning)하는 기법이다. 노브는 데이터베이스에 설정된 파라미터에 해당한다. 따라서, 이하 파라미터와 노브는 같은 의미로 사용한다.

【0017】 연구자는 RocksDB를 중심으로 노브(파라미터) 최적화 기법을 개발하였다. 따라서, 연구자는 RocksDB를 기준으로 모델을 구축하였고 성능을 검증하였다. 다만, 이하 설명하는 기술이 특정 유형의 데이터베이스에 한정되는 것은 아니다.

【0018】 RocksDB는 LSM-트리(log-structured merge-tree) 기반 키-값 데이터베이스이다. RocksDB에서 주요 구성은 멤테이블(memtable), SST 파일 및 컴팩션(compaction)이다. 이와 같은 노브는 RocksDB의 데이터 로딩 성능에 영향을 준다.

【0019】 이하 설명에서 두 개의 메트릭스(metrics)를 사용한다. 메트릭스는 업무 수행 결과를 보여주는 계량적 분석을 의미한다. 두 개의 메트릭스는 내부 메트릭스 및 외부 메트릭스이다. RocksDB 벤치마크는 약 400개의 통계적 정보들을 포함하는 데이터베이스의 상태를 기록한다. 연구자는 이 데이터베이스의 상태를 전처리하여 148개의 통계적 정보를 내부 메트릭스로 구성하였다. 내부 메트릭스는 블록 캐시 히트 수(block cache hit count), 블록 리드(block reads), 멤테이블 히트 수(memtable hit count) 등을 포함할 수 있다. 외부 메트릭스는 데이터베이스의 성능을 나타내는 지표들로 구성된다. 외부 메트릭스는 실행 시간, 쓰기 증폭 요인 등을 포함할 수 있다.

【0021】 이하 파라미터 최적화 과정은 컴퓨터 장치에서 수행된다고 설명한다. 컴퓨터 장치는 데이터 처리, 학습 모델 수행 등이 가능한 장치이다. 컴퓨터 장치는 PC, 스마트기기, 네트워크의 서버, 데이터 처리 전용 칩셋 등의 형태를 가질 수 있다.

【0023】 도 1은 데이터베이스의 최적 파라미터를 예측하는 개략적인 과정(100)의 예이다.

【0024】 먼저, 컴퓨터 장치는 타깃 워크로드의 정보를 입력받는다(110). 타깃 워크로드는 현재 데이터베이스에서 수행되는 워크로드이다. 현재 데이터베이스

가 최적화 대상이다.

【0025】 최적 파라미터 예측은 후술한 학습모델을 통해 수행된다. 컴퓨터 장치는 워크로드별로 모델을 구축하고, 구축된 모델을 이용하여 현재 데이터베이스의 노브를 최적화한다. 다만 타깃 워크로드를 학습하기 위한 샘플이 충분하지 않을 수 있다. 따라서, 컴퓨터 장치는 데이터베이스의 최적화 모델의 학습을 위한 샘플들을 선별한다(120). 컴퓨터 장치는 종전에 마련된 다양한 워크로드들 중 타깃 워크로드와 유사한 워크로드를 매핑한다(120).

【0026】 연구자는 사전에 마련된 다양한 워크로드들의 샘플들 중 타깃 워크로드와 유사한 워크로드를 선별하여 이용하였다. 워크로드의 유사도는 타깃 워크로드와 사전에 마련된 워크로드 샘플들 사이의 유클리디언 거리(Euclidean distance)로 평가할 수 있다. 컴퓨터 장치는 워크로드의 내부 메트릭스를 기준으로 유사도를 평가한다. 컴퓨터 장치는 내부 메트릭스의 값으로 결정되는 유클리디언 거리를 기준으로 워크로드들 사이의 유사도를 평가할 수 있다. 컴퓨터 장치는 사전에 마련된 다양한 워크로드들 중 타깃 워크로드와의 유클리디언 거리가 가장 짧은 워크로드(유사 워크로드)를 후보로 선별할 수 있다. 컴퓨터 장치는 유사 워크로드의 다수 샘플을 선별할 수 있다. 컴퓨터 장치는 10개 이상의 샘플을 이용할 수 있다. 연구자는 실험적으로 10개의 샘플들을 학습 데이터로 사용하였다.

【0027】 컴퓨터 장치는 유전 알고리즘으로 노브 최적값을 결정한다. 이때 노브는 희소 벡터 형태로 표현된다. 이를 위하여 컴퓨터 장치는 룩-업 테이블(lookup table)을 학습하는 신경망을 이용한다. 컴퓨터 장치는 노브의 희소 메트릭스를

생성하는 신경망을 학습한다(130). 컴퓨터 장치는 노브에 대한 희소 벡터를 생성하고(131), 룩-업 테이블 학습을 이용하여 신경망을 학습한다(132). 컴퓨터 장치는 타깃 워크로드에 유사한 워크로드의 설정 매트릭스를 산출할 수 있다. 구체적인 학습 과정은 후술한다.

【0028】 컴퓨터 장치는 유전 알고리즘으로 노브 최적값을 결정한다(140). 유전 알고리즘에서 적합도 값은 레퍼런스값(디폴트 값)과 예측값을 비교하여 결정된다. 한편, 예측값은 예측 모델을 통해 산출된다. 이를 위하여 컴퓨터 장치는 예측 모델을 구축한다(141). 예측 모델은 GRU 기반의 모델로 외부 매트릭스를 예측한다. 예측 모델에 입력되는 노브의 특징 벡터는 130과정에서 구축한 모델을 통해 산출될 수 있다. 구체적인 과정은 후술한다. 이후 컴퓨터 장치는 유전 알고리즘을 이용하여 현재 데이터베이스의 최적 설정값을 추천한다(142).

【0030】 연구자는 노브(노브)들의 특징(feature)을 학습하기 위한 신경망을 고안하였다. 연구자는 자연어 처리에서 이용되는 Word2Vec과 같이 노브들의 특징을 학습하는 신경망을 구축하였다. 노브 특징은 벡터들은 워크로드 특이적 특징을 포함하고, 각 노브의 값을 설명해야 한다. 도 2는 데이터베이스의 노브 특징을 학습하는 과정에 대한 예이다. 도 2는 전술한 룩-업 테이블을 학습하는 신경망을 구축하는 과정이다. 도 2는 특정한 타깃 워크로드를 대상으로 한다고 가정한다.

【0031】 도 2(A)는 노브들의 값에 대한 희소 벡터를 생성하는 과정의 예이다. 컴퓨터 장치는 모든 노브들의 원-핫 벡터들(one-hot vectors)을 결합(concatenation)하여 희소 벡터들을 생성한다.

【0032】 원-핫 벡터들의 크기는 노브들에서 특정 값들을 카운트한 값이다. 원-핫 벡터들의 크기는 노브마다 서로 다를 수 있다. 예컨대, 노브  $i$  (knob $_i$ )의 범위가  $[1,6]$ 이고 그 값이 2라면, knob $_i$ 의 원-핫 벡터  $c_i$ 는  $[0,1,0,0,0,0]$ 일 수 있다. 희소 벡터  $s \in \mathbb{R}^{d_s}$ 는 아래 수학식 1과 같이 주어진 설정에 따라  $n$ 개의 노브들의 원-핫 벡터들이 연속적으로 결합되어 생성될 수 있다.

#### 【0034】 【수학식 1】

$$s = [c_1; c_2; \dots; c_n]$$

【0036】 컴퓨터 장치는 모든 학습 샘플들의 설정들에 대한 희소 벡터들을 생성한다. 이후 컴퓨터 장치는 희소 벡터들과 내부 메트릭스(internal metric)들을 이용하여 신경망(neural network)을 학습시킨다. 신경망은 룩-업 테이블(look-up table)을 학습하게 된다. 룩-업 테이블을 학습하는 과정은 연속적인 데이터로 상호간 연관 관계 있는 자연어 문장을 처리하는 학습 모델(Yoshua Bengio et al., A Neural Probabilistic Language Model, The Journal of Machine Learning Research

Volume 3, 2003)의 학습 과정과 유사하다. 해당 학습 모델은 직전에 입력되는  $n-1$ 개 단어들로 다음 단어를 예측하는  $n$ -gram 언어 모델이다.

【0037】 도 2(B)는 록-업 테이블의 학습 과정의 예이다. 신경망은 희소 벡터들과 내부 매트릭스들 사이의 관계를 학습한다. 이를 통해 신경망은 워크로드의 특징을 포함하는 가중치들을 갖게 된다.

【0038】 신경망은 선형 계층에 희소 벡터들이 임베딩되고, 신경망은 아래 수학적 식 2와 같이 매핑과정에서 획득한 유사한 워크로드의 내부 매트릭스  $v$ 를 예측하게 된다. 신경망은 록-업 테이블을 산출하게 된다. 또한, 신경망은 노브들의 희소 벡터들을 입력받아 타깃 워크로드에 대응하는 내부 매트릭스를 출력한다.

#### 【0040】 【수학적 식 2】

$$v = (sW_s + b_s)W_v + b_v$$

【0041】  $v \in \mathbb{R}^{d_v}$  는 유사한 워크로드의 내부 매트릭스들이다.  $W_s \in \mathbb{R}^{d_s \times d_x}$  록-업 테이블이고,  $W_v \in \mathbb{R}^{d_x \times d_v}$  는 학습가능한 가중치들이다.

【0043】 록-업 테이블은 주어진 노브들의 특징 벡터를 선택하는데 이용된다. 도 2(C)는 록-업 테이블을 통해 노브의 특징 벡터를 산출하는 전체 과정의 예이다. 특징 벡터(Knob2vec)는 록-업 테이블로부터 선택된다. 특징 벡터는 희소 벡터들에

서 선택된 하나의 위치에 해당한다. 만약 knob<sub>1</sub>에서 1의 위치가  $\alpha$  라면, knob<sub>1</sub>의 특징 벡터는 룩-업 테이블에서  $\alpha$  번째 행(row)이다. 모든 노브들로부터 특징 벡터를 획득한 후 컴퓨터 장치는 벡터들을 결합하여 설정 매트릭스(configuration matrix)  $X \in \mathbb{R}^{d_k \times d_x}$  을 구성한다.

【0045】 컴퓨터 장치는 유전 알고리즘을 사용하여 데이터베이스의 노브 설정을 추천할 수 있다. 유전 알고리즘은 해집단(population)에서 목적 함수를 달성하는 가장 좋은 해를 찾는 알고리즘이다. 아래 표 1은 노브 최적화를 위한 수도코드(pseudo-code)의 예이다.

【0046】 먼저, 컴퓨터 장치는  $n$ 개의 염색체들을 갖는 해집단  $Y_{init}$ 을 생성한다(라인 2). 염색체(chromosome)들은 노브 최적화를 위한 주어진 설정들에서의 노브들이다. 염색체들을 구성하는 노브들의 설정값은 임의의 값으로 생성될 수 있다.

【0047】 컴퓨터 장치는 적합도 함수를 이용하여 각 염색체의 적합도(fitness value)를 연산한다( $f=M(Y_{init})$ , 라인 5). 적합도 함수  $M()$ 은 아래 설명할 예측 모델로 구현한다. 컴퓨터 장치는 적합도 값을 정렬하여 상위 절반의 염색체들  $Y_{best}$ 을 선택하고, 선택한 집단에 대한 교차(crossover)를 수행한다(라인 6-7). 컴퓨터 장치는 새로운  $n/2$ 개 염색체들  $Y_{half}$ 을 생성하여  $Y_{best}$ 과 변이(mutation)를 수행한다(라인 8-9). 마지막으로 컴퓨터 장치는 선택한 염색체들  $Y_{best}$ 과 변이로 생성된 값을 결합한다(라인 10). 컴퓨터 장치는 라인 5~10의 과정을  $g$ 번 반복하고, 최종적으로 최적

해를 산출한다.

【0049】 【표 1】

Algorithm 1 GA in Knob Tuning	
<b>Input:</b>	Population size $n$ , Number of Generation $g$ , Predictive model $M$
<b>Output:</b>	Best solution $Y_b$
1:	<b>begin</b>
2:	Generation initial population of $n$ chromosomes $Y_{init}$
3:	Set iteration counter $cnt = 0$
4:	<b>for</b> $cnt < g$ <b>do</b>
5:	Compute the fitness value of each chromosomes: $f = M(Y_{init})$
6:	Select half of the top fitness value, $Y_{best}$
7:	Apply crossover operation on $Y_{best}$
8:	Generate half of $n$ chromosomes $Y_{half}$
9:	Apply mutation on crossovered $Y_{best}$ with $Y_{half}$
10:	Combine old $Y_{best}$ and newly generated $Y_{best}$ : $Y_{half} = [old Y_{best}; new Y_{best}]$
11:	<b>end for</b>
12:	Return the best solution, $Y_{best}$
13:	<b>end</b>

【0051】 유전 알고리즘에서 선택 과정은 적합도 값을 정렬하여 비교하는 과정을 포함한다. 적합도 값은 4개의 외부 매트릭스들로 정의된다. 외부 매트릭스들은 실행 시간(execution time, TIME), 데이터 처리율(data processing rate, RATE), 쓰기 증폭 인자(write amplification factor, WAF) 및 공간 증폭(space amplification, SA)에 대한 매트릭스로 구성된다. 실행 시간(TIME)은 데이터 처리의 시작과 종료 사이의 시간 간격이다. 데이터 처리율(RATE)은 초(시간) 당 처리된 데이터 크기이다. 쓰기 증폭 인자(WAF)는 물리적 데이터 크기 및 논리적 데이터 크기의 비율로서 RocksDB의 사용 중 부가적인 데이터 쓰기의 양을 의미한다. 쓰기 증

폭 인자(WAF)는 낮을 수록 높은 성능을 의미한다. 공간 증폭(SA)은 실제 LSM-트리에서 기록된 데이터의 크기로 로드된 데이터 크기를 측정한 값이다. 공간 증폭(SA)은 실제 물리적 공간에 대한 성능 지표이며 낮을 수록 높은 성능을 의미한다.

【0052】 컴퓨터 장치는 예측 값과 디폴트(default) 값 사이의 절대 오류(absolute error, AE)의 합을 기준으로 적합도 값을 비교한다. 디폴트 값은 타깃 위크로드에서 디폴트 설정의 결과에 해당한다. 절대 오류(AE)는 아래 수학적 식 3과 같이 연산될 수 있다.

【0054】 【수학적 식 3】

$$AE = \sum |\hat{y} - y|$$

【0055】  $\hat{y}$ 는 예측 값이고,  $y$ 는 디폴트 값이다. 예측 값은 별도의 예측 모델을 이용하여 예측된다.

【0057】 연구자는 유전 알고리즘에서 적합도 함수로 전술한 외부 매트릭스들을 예측하는 예측 모델을 이용하였다. 예측 모델은 주어진 노브 특징 벡터들로부터 외부 매트릭스들을 예측한다. 연구자는 어텐션(attention) 메커니즘이 적용된 GRU(Gated Recurrent Unit)를 이용하여 예측 모델을 구현하였다. 예측 모델은 인코더 및 디코더를 포함한다. 인코더는 노브 특징 벡터들을 입력받는다. GRU를 갖는

인코더는 아래 수학적 식 4 및 5에 따른 동작을 수행한다.

【0059】 【수학적 식 4】

$$\tilde{x} = XW_x + b_x$$

【0060】 【수학적 식 5】

$$o^e, h^e = GRU(\tilde{x})$$

【0061】  $X$ 는 도 2에서 설명한 신경망을 이용하여 산출하는 설정 매트릭스이다.  $W_x \in \mathbb{R}^{d_x \times d_h}$ 는 학습가능한 가중치이다.  $\tilde{x}$ 는 선형 투사의 출력이다.  $o^e \in \mathbb{R}^{d_k \times d_h}$  및  $h^e \in \mathbb{R}^{d_h}$ 는 GRU 인코더의 출력이다.

【0063】 예측 모델의 디코더는 아래 수학적 식 6 및 7에 따른 동작을 수행한다.  
디코더의 GRU는 인코더의 출력을 입력으로 받는다.

【0065】 【수학적 식 6】

$$o_t^d, h_t^d = GRU(o_{t-1}^d, h_{t-1}^d)$$

## 【0066】 【수학식 7】

$$a_t = \text{softmax}(o_t^d W_a o^e)$$

【0068】  $t$ 는 시간이다.  $o^d \in \mathbb{R}^{d_z \times d_h}$  및  $h^d \in \mathbb{R}^{d_h}$ 는 디코더의 출력들이다.  $W_x \in \mathbb{R}^{d_h}$ 는 학습가능한 가중치이다.  $a_t \in \mathbb{R}^{d_k}$ 는 어텐션 점수이다. 어텐션 점수는 인코더의 출력과 디코더의 출력을 곱하여 연산된다. 컨텍스트(context) 벡터는 디코더의 출력과 어텐션 점수를 곱하여 산출된다. 마지막으로 전연결 계층(fully connected layer)는 컨텍스트 벡터를 입력받아 예측 값  $\hat{y}$ 을 출력한다.

【0070】 이하 연구자가 연구한 노브 최적화 기법의 성능을 검증한 결과를 설명한다.

【0071】 연구자는 RocksDB 벤치마크 툴을 이용하여 모델 구축을 위한 샘플들을 생성하였다. 연구자는 충분한 학습 샘플들을 생성하기 위하여 16개의 다양한 워크로드들에서 20,000 개의 설정값을 임의로 추출하였다. 학습 및 검증을 위한 워크로드 샘플들은 엔트리의 개수, 읽기-쓰기 비율 및 업데이트 옵션을 기준으로 서로 다른 크기들로 구성되었다. 검증용 타겟 워크로드는 학습 과정에서 사용되지 않은 다른 옵션으로 설정된 워크로드를 선별하였다. 아래 표 2는 학습 및 검증에 사용된 워크로드들의 예이다. 전체 데이터 크기(data size)는 1GB로 설정되었고, 키의 크

기(key size)는 16B로 설정되었다. 엔트리들의 개수는  $\frac{\text{data size}}{(\text{key size} + \text{value size})}$ 로 연산된다.

업데이트 옵션은 읽기, 쓰기 및 수정을 포함한다.

### 【0073】 【표 2】

TRAINING (학습용 워크로드)					
Index	Value size (B)	엔트리 개수	READ	WRITE	UPDATE
0	1024	1032444	90%	10%	X
1	4096	261124			
2	16384	65472			
3	65536	16380			
4	1024	1032444	50%	50%	X
5	4096	261124			
6	16384	65472			
7	65536	16380			
8	1024	1032444	10%	90%	X
9	4096	261124			
10	16384	65472			
11	65536	16380			
12	1024	1032444	-	-	O
13	4096	261124			
14	16384	65472			
15	65536	16380			
TESTING (검증용 워크로드)					
Name	Value size (B)	# 엔트리 개수	READ	WRITE	UPDATE
R70W30_8	8192	130816	70%	30%	X
R30W70_8			30%	70%	X
U_8			-	-	O
R70W30_32	32768	32752	70%	30%	X
R30W70_32			30%	70%	X
U_32			-	-	O

【0075】 외부 메트릭스들은 실행 시간(TIME), 데이터 처리율(RATE), 쓰기 증폭 인자(WAF) 및 공간 증폭(SA)에 대한 메트릭스를 이용하였다. 연구자는 RocksDB

의 최적 설정을 자동으로 추천하는 능력을 평가하였다. 연구자는 아래 수학적 식 8과 같이 외부 메트릭스들로 연산되는 점수(score)를 연산하였다.

**【0077】 【수학적 식 8】**

$$score = \log\left(\frac{TIME_{def}}{TIME}\right) + \log\left(\frac{RATE}{RATE_{def}}\right) + \log\left(\frac{WAF_{def}}{WAF}\right) + \log\left(\frac{SA_{def}}{SA}\right)$$

**【0078】** 수학적 식 8에서 def는 디폴트 외부 메트릭스들을 의미한다. 상기 점수는 예측된 외부 메트릭스를 디폴트 외부 메트릭스들로 나눈 값을 항목으로 갖는다. 디폴트 값은 타깃 워크로드에서 디폴트 설정값이다. 즉, 상기 점수는 성능이 향상된 정도를 나타낸다.

**【0080】** 연구자는 적절한 유사 워크로드를 찾기 위하여 사전에 획득한 16개의 학습용 워크로드 샘플들의 개수를 조절하였다. 연구자는 샘플들의 개수를 [10, 50, 100, 1000]으로 설정하고 학습용 워크로드들 중 16번째의 워크로드를 타깃 워크로드로 설정하였다. 워크로드의 유사도는 유클리디언 거리를 기준으로 측정하였다.

**【0081】** 도 3은 다양한 개수의 샘플 모집단을 이용하여 타깃 워크로드와 유사한 워크로드를 찾은 결과이다. 도 3을 살펴보면 타깃 워크로드와 유사한 워크로드는 12번째 워크로드로 나타난다. 서로 다른 개수의 샘플들에서 모두 동일한 결과

를 보였다. 따라서, 도 3의 결과에 따르면 타깃 워크로드와 유사한 워크로드를 찾기 위한 샘플들의 개수는 10개면 충분하다고 할 수 있다.

【0083】 연구자는 종래 노브 최적화 기법과 제안한 최적화 기법의 성능을 비교하였다. 제안한 노브 최적화 기법은 전술한 바와 같은 유전 알고리즘을 사용하는 기법이다. 종래 노브 최적화 기법은 OtterTune, CDBTune, 페이스북의 최적화 노브 설정(Facebook) 및 DBA를 사용하였다. 도 4는 노브 최적화 기법의 성능을 평가한 결과이다. 성능은 수학적 8에서 설명한 점수로 평가하였다. 도 4에서 K2vTune이 연구자가 제안한 최적화 기법이다. 도 4(a) ~ (f)는 각각 검증용 워크로드 R70W30 8, R30W70 8, U 8, R70W30 32, R30W70 32 및 U 32에 대한 결과를 나타낸다. 모든 검증 워크로드에서 제안한 노브 최적화 기법(K2vTune)의 성능이 높은 것을 알 수 있다.

【0085】 전술한 바와 같이 유전 알고리즘에서 적합도 함수는 예측 모델과 디폴트 값의 차이를 기준으로 결정된다. 따라서, 해당 예측 모델의 정확도가 노브 최적화 결과에 영향을 미친다. 연구자는 형태의 예측 모델을 구축하고, 각 모델의 성능을 평가하였다. 연구자는 단일 신경망, GRU, 양방향(bidirectional) GRU(BiGRU), 어텐션을 갖는 GRU(GRU+Attn) 및 어텐션을 갖는 양방향 GRU 형태로 각각 예측 모델(BiGRU+Attn)을 구현하였다. 연구자는 K2vTune을 이용하지 않고 단순히 노브 특징을 산출하는 단일 신경망(Single)과 K2vTune을 이용하는 단일 신경망(Single+Knob2vec)에 대한 성능을 모두 평가하였다. 도 5는 예측 모델의 유형에 따

른 노브 최적화 성능을 비교한 결과이다. 도 5(a) ~ (f)는 각각 검증용 워크로드 R70W30 8, R30W70 8, U 8, R70W30 32, R30W70 32 및 U 32에 대한 결과를 나타낸다. 도 5를 살펴보면 K2vTune을 이용하는 모델이 K2vTune을 이용하지 않는 모델보다 성능이 높았다. 도 5를 살펴보면, 단일 신경망보다 GRU가 성능이 높았고, 어텐션을 사용하는 모델이 어텐션을 사용하지 않는 모델보다 성능이 높았다.

【0090】 도 6은 노브 최적화를 추천하는 컴퓨터 장치(200)에 대한 예이다. 컴퓨터 장치(200)는 저장장치(210), 메모리(220), 연산장치(230), 인터페이스 장치(240) 및 통신장치(250)를 포함할 수 있다.

【0091】 저장장치(210)는 RocksDB의 최적 파라미터(노브)를 결정하는 과정에서 사용되는 학습모델을 저장한다. 저장장치(210)는 노브들의 회소 벡터를 입력받아 데이터베이스의 내부 메트릭스를 산출하는 제1 신경망을 저장한다. 저장장치(210)는 노브들의 설정 메트릭스를 입력받아 데이터베이스의 외부 메트릭스를 산출하는 제2 신경망을 저장한다.

【0092】 전술한 바와 같이, 내부 메트릭스는 RocksDB의 상태를 나타내는 항목으로 구성된다. 내부 메트릭스는 블록 캐시 히트 수, 블록 리드 및 멤테이블 히

트 수를 포함할 수 있다. 외부 메트릭스는 RocksDB의 성능을 나타내는 항목으로 구성될 수 있다. 외부 메트릭스는 실행 시간, 데이터 처리율, 쓰기 증폭 인자 및 공간 증폭의 항목을 포함할 수 있다.

【0093】 저장장치(210)는 제1 신경망 및 제2 신경망을 학습하는 과정을 제어하는 프로그램(코드)을 저장할 수 있다.

【0094】 저장장치(210)는 RocksDB의 최적화 과정을 제어하는 프로그램을 저장할 수 있다.

【0095】 메모리(220)는 RocksDB의 최적화 과정을 제어하는 과정에서 생성되는 데이터 및 정보 등을 저장할 수 있다.

【0096】 인터페이스 장치(240)는 외부로부터 일정한 명령 및 데이터를 입력받는 장치이다. 인터페이스 장치(240)는 물리적으로 연결된 입력 장치 또는 외부 저장장치로부터 RocksDB의 타깃 워크로드의 정보를 입력받을 수 있다. 인터페이스 장치(240)는 RocksDB의 최적 파라미터를 외부 객체에 전달할 수도 있다.

【0097】 통신장치(250)는 유선 또는 무선 네트워크를 통해 일정한 정보를 수신하고 전송하는 구성을 의미한다. 통신장치(250)는 외부 객체로부터 RocksDB의 타깃 워크로드의 정보를 수신할 수 있다. 또는 통신장치(250)는 RocksDB의 최적 파라미터를 사용자 단말과 같은 외부 객체에 송신할 수도 있다.

【0098】 인터페이스 장치(240)는 통신장치(250)를 통해 수신된 정보를 내부로 전달하는 구성일 수 있다.

【0099】연산 장치(230)는 전술한 제1 신경망 및 제2 신경망을 구축하는 과정을 수행할 수 있다. 이 과정에서 연산 장치(230)는 타깃 워크로드와 유사한 워크로드를 선별할 수 있다. 연산 장치(230)는 사전에 수집한 다양한 워크로드들 중 타깃 워크로드와 가장 유사한 워크로드를 후보로 선택할 수 있다. 연산 장치(230)는 워크로드의 내부 메트릭스를 기준으로 다양한 워크로드들과 타깃 워크로드의 유클리디언 거리를 연산하고, 해당 거리가 가장 짧은 워크로드를 후보로 선별할 수 있다. 연산 장치(230)는 후보로 선별한 워크로드의 샘플들을 이용하여 제1 신경망 및 제2 신경망의 학습을 수행할 수 있다. 제1 신경망 및 제2 신경망 각각의 구조 및 학습과정은 전술한 바와 같다.

【0100】연산 장치(230)는 유전 알고리즘을 이용하여 RocksDB의 최적 설정값을 결정할 수 있다.

【0101】연산 장치(230)는 유전 알고리즘에서 해집합을 생성하고 최적해를 산출한다. 이 과정에서 연산 장치(230)는 염색체(노브들)에 대한 희소 벡터를 산출하고, 도 2에서 설명한 제1 신경망에 입력하여 내부 메트릭스를 생성할 수 있다. 연산 장치(230)는 룩-업 테이블을 이용하여 전체 노브들의 설정 메트릭스  $X$ 를 생성한다.

【0102】연산 장치(230)는 표 1에서 설명한 유전 알고리즘을 이용하여 최적 설정값을 결정할 수 있다. 연산 장치(230)는 해당 유전 알고리즘에서 사용되는 적합도(수학식 3 참조)를 연산할 수 있다. 연산 장치(230)는 설정 메트릭스를 입력받아 외부 메트릭스를 예측하는 GRU 기반의 제2 신경망을 이용하여 적합도를 연산할

수 있다.

【0103】 전술한 바와 같이 제2 신경망은 설정 매트릭스를 입력받는 인코더, 인코더의 출력값을 입력받는 디코더 및 디코더의 출력과 어텐션 점수로 산출되는 컨텍스트 벡터를 입력받아 예측값(외부 매트릭스)을 출력하는 전연결 계층을 포함한다.

【0104】 연산 장치(230)는 데이터를 처리하고, 일정한 연산을 처리하는 프로세서, AP, 프로그램이 임베디드된 칩과 같은 장치일 수 있다.

【0106】 또한, 상술한 바와 같은 데이터베이스 최적화 방법 및 RocksDB의 노브 최적화 방법은 컴퓨터에서 실행될 수 있는 실행가능한 알고리즘을 포함하는 프로그램(또는 어플리케이션)으로 구현될 수 있다. 상기 프로그램은 일시적 또는 비일시적 판독 가능 매체(non-transitory computer readable medium)에 저장되어 제공될 수 있다.

【0107】 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM (read-only memory), PROM (programmable read only memory), EPROM(Erasable PROM, EPROM) 또는 EEPROM(Electrically EPROM) 또는 플래시 메모

리 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.

【0108】 일시적 판독 가능 매체는 스테틱 램(Static RAM, SRAM), 다이내믹 램(Dynamic RAM, DRAM), 싱크로너스 디램 (Synchronous DRAM, SDRAM), 2배속 SDRAM(Double Data Rate SDRAM, DDR SDRAM), 증강형 SDRAM(Enhanced SDRAM, ESDRAM), 동기화 DRAM(Synclink DRAM, SLDRAM) 및 직접 램버스 램(Direct Rambus RAM, DRRAM) 과 같은 다양한 RAM을 의미한다.

【0109】 본 실시예 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시례는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

## 【청구범위】

### 【청구항 1】

컴퓨터 장치는 데이터베이스의 타깃 워크로드의 정보를 입력받는 단계;

상기 컴퓨터 장치는 상기 타깃 워크로드에 대한 샘플들을 선별하는 단계; 및

상기 컴퓨터 장치는 유전 알고리즘을 이용하여 상기 데이터베이스에 대한 최적 설정값을 결정하는 단계를 포함하되,

상기 최적 설정값을 결정하는 단계에서 상기 컴퓨터 장치가 상기 유전 알고리즘에서 사용되는 염색체에 해당하는 노브들의 희소 벡터를 산출하고, 상기 희소 벡터를 입력받아 신경망이 산출하는 내부 메트릭스를 이용하여 설정 메트릭스하고, 상기 컴퓨터 장치가 상기 설정 메트릭스를 입력받는 학습모델이 출력하는 예측값 및 상기 타깃 워크로드의 디폴트값으로 결정되는 적합도를 기준으로 상기 최적 설정값을 결정하고,

상기 신경망은 상기 샘플들을 이용하여 학습되는 노브 벡터 표현을 이용한 데이터베이스의 최적화 방법.

### 【청구항 2】

제1항에 있어서,

상기 데이터베이스는 RocksDB인 노브 벡터 표현을 이용한 데이터베이스의 최적화 방법.

**【청구항 3】**

제1항에 있어서,

상기 컴퓨터 장치는 사전에 수집한 다양한 워크로드들 및 상기 타깃 워크로드 사이의 유클리디언 거리를 기준으로 상기 다양한 워크로드들 중 상기 타깃 워크로드와 가장 유사한 하나의 워크로드를 선별하는 단계를 더 포함하고, 상기 샘플들은 상기 하나의 워크로드에 대한 샘플들인 노브 벡터 표현을 이용한 데이터베이스의 최적화 방법.

**【청구항 4】**

제3항에 있어서,

상기 유클리디언 거리는 워크로드의 내부 메트릭스를 기준으로 결정되고,

상기 내부 메트릭스는 블록 캐시 히트 수, 블록 리드 및 메모이저블 히트 수의 항목을 포함하는 노브 벡터 표현을 이용한 데이터베이스의 최적화 방법.

**【청구항 5】**

제1항에 있어서,

상기 학습모델은 GRU(Gated Recurrent Unit) 기반의 모델로서,

상기 설정 메트릭스를 입력받는 인코더;

상기 인코더의 출력값을 입력받는 디코더; 및

상기 디코더의 출력과 어텐션 점수로 산출되는 컨텍스트 벡터를 입력받아 상기 예측값을 출력하는 전연결 계층을 포함하는 노브 벡터 표현을 이용한 데이터베이스

이스의 최적화 방법.

### 【청구항 6】

제1항에 있어서,

상기 예측값은 상기 데이터베이스의 외부 메트릭스를 포함하고,

상기 외부 메트릭스는 실행 시간, 데이터 처리율, 쓰기 증폭 인자 및 공간 증폭의 항목을 포함하는 노브 벡터 표현을 이용한 데이터베이스의 최적화 방법.

### 【청구항 7】

RocksDB의 타깃 워크로드의 정보를 입력받는 인터페이스 장치;

노브들의 회소 벡터를 입력받아 데이터베이스의 내부 메트릭스를 산출하는 제1 신경망 및 노브들의 설정 메트릭스를 입력받아 데이터베이스의 외부 메트릭스를 산출하는 제2 신경망을 저장하는 저장장치; 및

유전 알고리즘을 이용하여 상기 RocksDB의 최적 설정값을 결정하는 연산장치를 포함하되,

상기 연산장치는 상기 유전 알고리즘에서 사용되는 염색체에 해당하는 노브들의 회소 벡터를 산출하고, 상기 회소 벡터를 상기 제1 신경망에 입력하여 산출되는 내부 메트릭스를 이용하여 설정 메트릭스를 산출하고, 상기 설정 메트릭스를 상기 제2 신경망에 입력하여 출력되는 외부 메트릭스 및 상기 타깃 워크로드의 디폴트값으로 결정되는 적합도를 기준으로 상기 최적 설정값을 결정하고,

상기 내부 메트릭스는 RocksDB의 상태를 나타내는 항목으로 구성되고, 상기

외부 메트릭스는 RocksDB의 성능을 나타내는 항목으로 구성되는 데이터베이스의 최적 파라미터를 결정하는 컴퓨터 장치.

#### 【청구항 8】

제7항에 있어서,

상기 내부 메트릭스는 블록 캐시 히트 수, 블록 리드 및 메모이블 히트 수를 포함하고,

상기 외부 메트릭스는 실행 시간, 데이터 처리율, 쓰기 증폭 인자 및 공간 증폭의 항목을 포함하는 데이터베이스의 최적 파라미터를 결정하는 컴퓨터 장치.

#### 【청구항 9】

제7항에 있어서,

상기 제1 신경망 및 상기 제2 신경망은 상기 타깃 워크로드와 유사한 워크로드의 샘플들을 이용하여 학습되고,

상기 유사한 워크로드는 사전에 수집한 워크로드들의 내부 메트릭스와 상기 타깃 워크로드의 내부 메트릭스의 유사도를 비교하여 결정되는 데이터베이스의 최적 파라미터를 결정하는 컴퓨터 장치.

#### 【청구항 10】

제7항에 있어서,

상기 제2 신경망은 GRU(Gated Recurrent Unit) 기반의 모델로서,

상기 설정 메트릭스를 입력받는 인코더;

상기 인코더의 출력값을 입력받는 디코더; 및

상기 디코더의 출력과 어텐션 점수로 산출되는 컨텍스트 벡터를 입력받아 상기 예측값을 출력하는 전연결 계층을 포함하는 데이터베이스의 최적 파라미터를 결정하는 컴퓨터 장치.

**【청구항 11】**

제7항에 있어서,

상기 적합도는 상기 외부 매트릭스와 상기 타깃 워크로드의 디폴트값의 차이의 합으로 결정되는 데이터베이스의 최적 파라미터를 결정하는 컴퓨터 장치.

**【요약서】****【요약】**

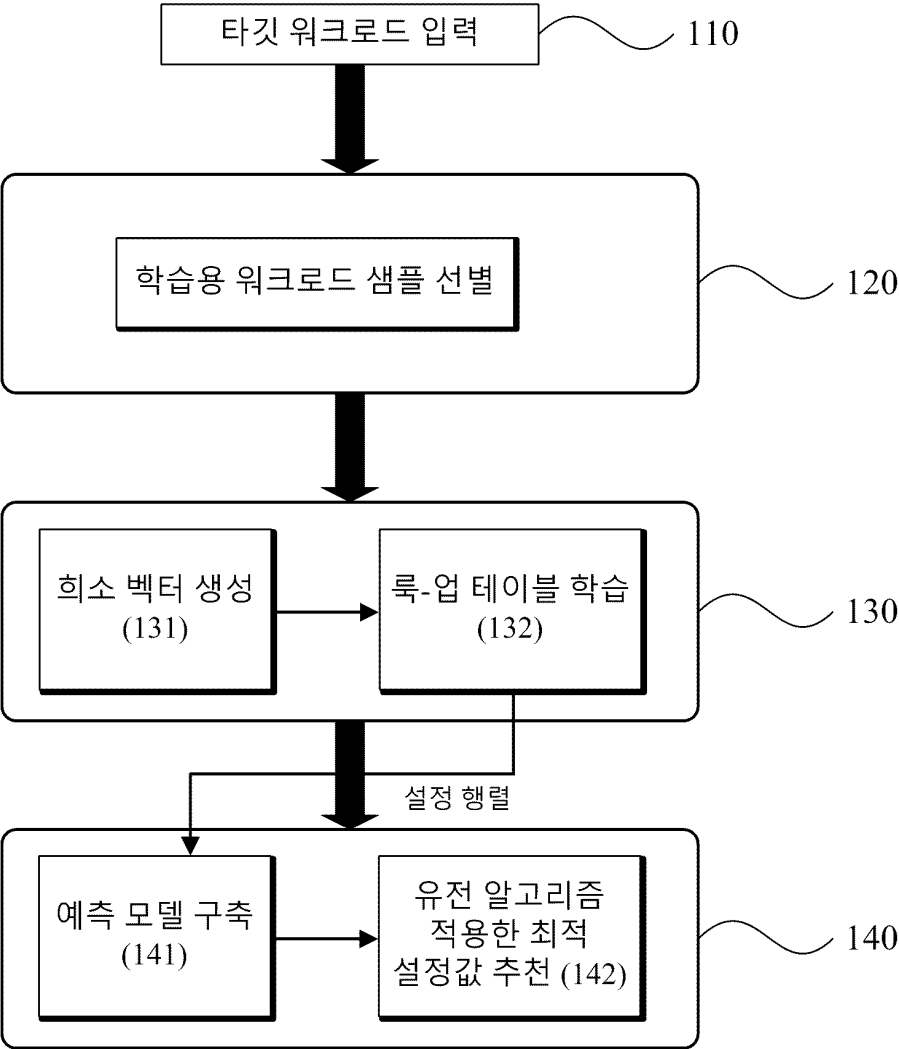
노브 벡터 표현을 이용한 데이터베이스의 최적화 방법은 컴퓨터 장치는 데이터베이스의 타겟 워크로드의 정보를 입력받는 단계 및 상기 컴퓨터 장치는 유전 알고리즘을 이용하여 상기 데이터베이스에 대한 최적 설정값을 결정하는 단계를 포함한다. 상기 컴퓨터 장치는 유전 알고리즘에서 노브들의 설정값을 입력받는 학습모델이 출력하는 예측값과 상기 타겟 워크로드의 디폴트값의 차이를 기준으로 적합도를 결정한다.

**【대표도】**

도 1

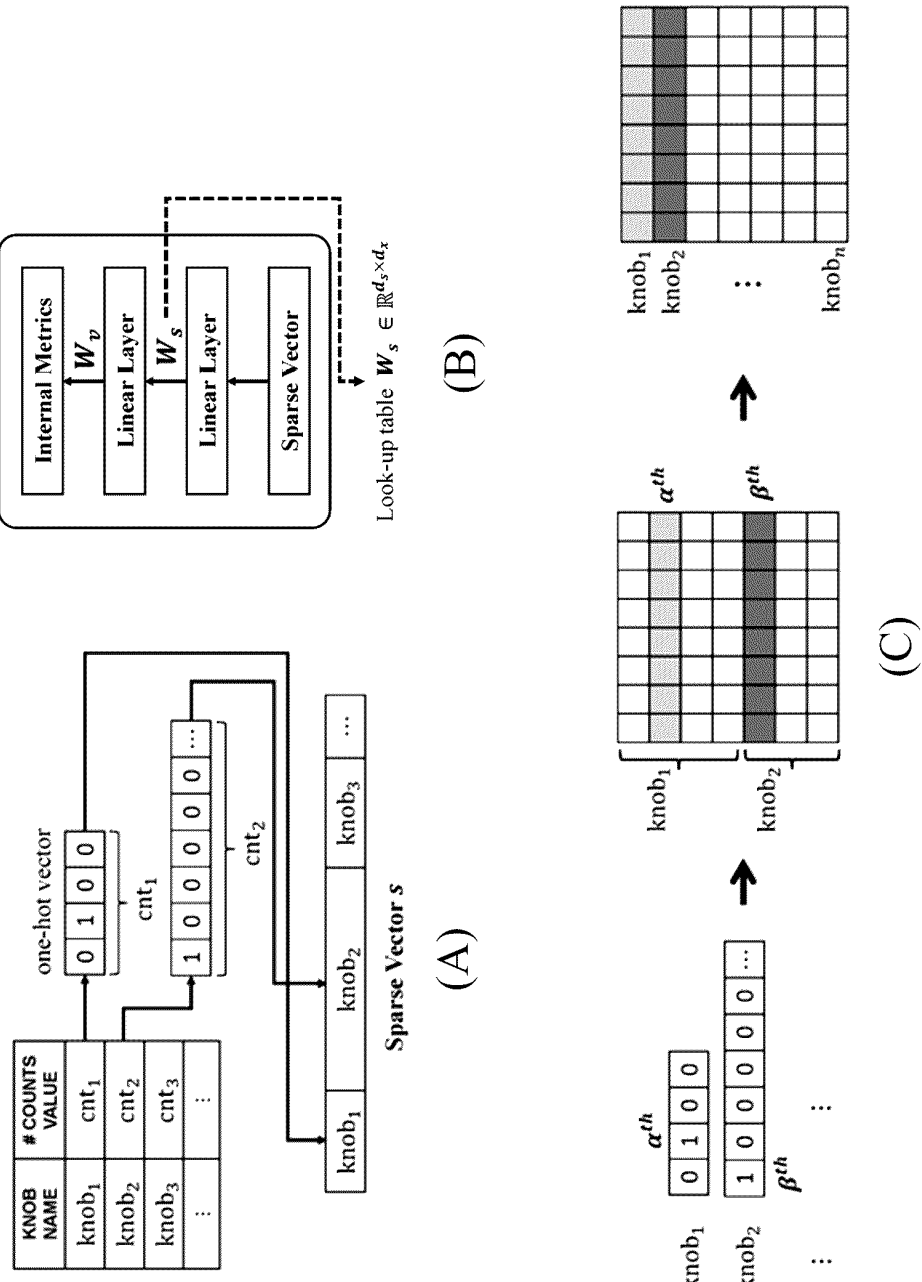
【도면】

【도 1】

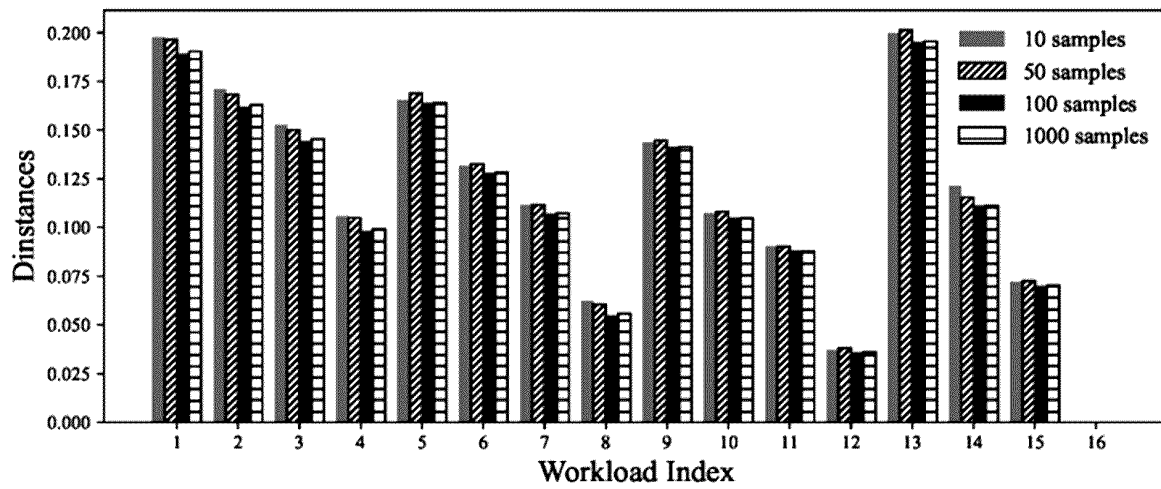


100

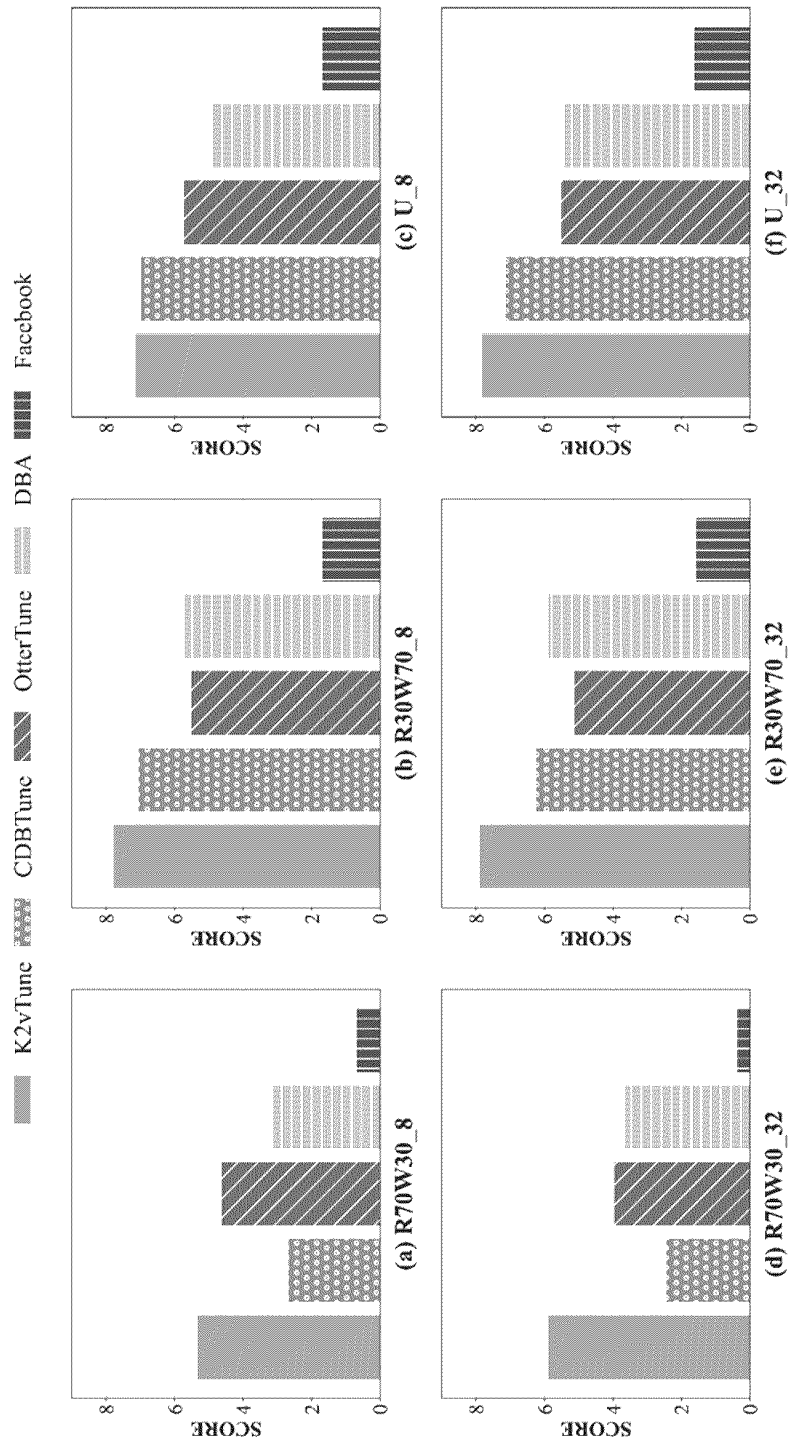
【図 2】



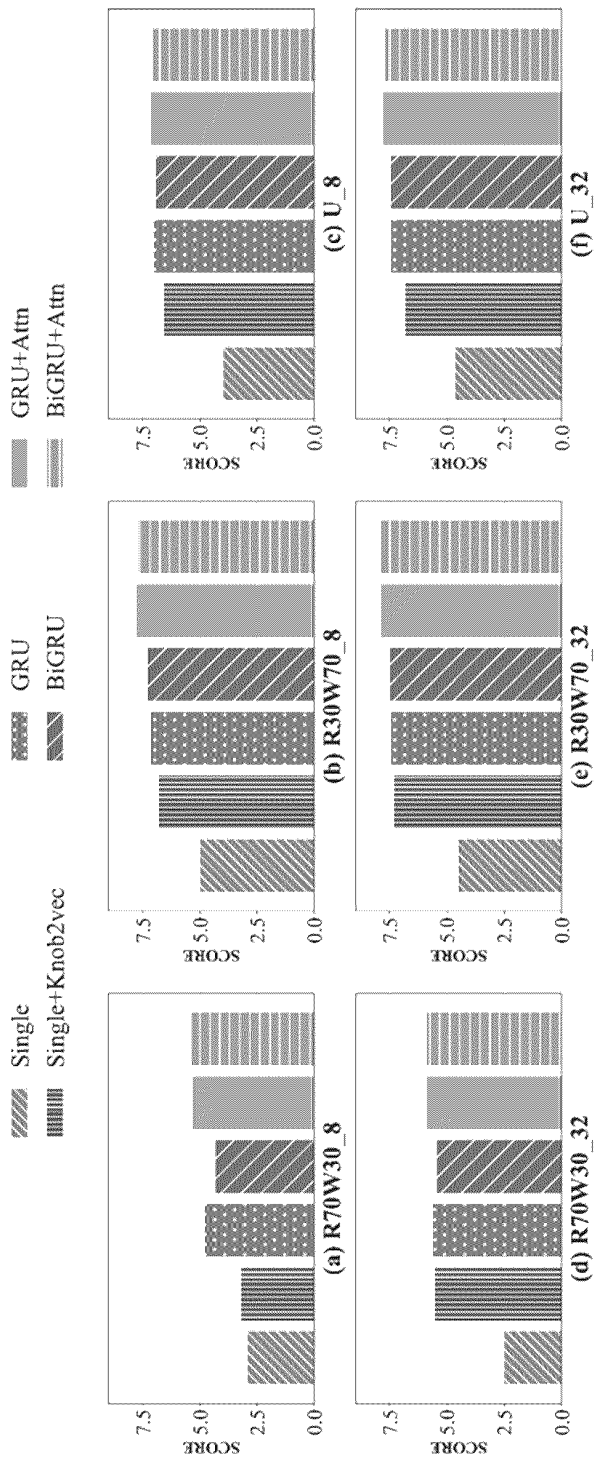
【도 3】



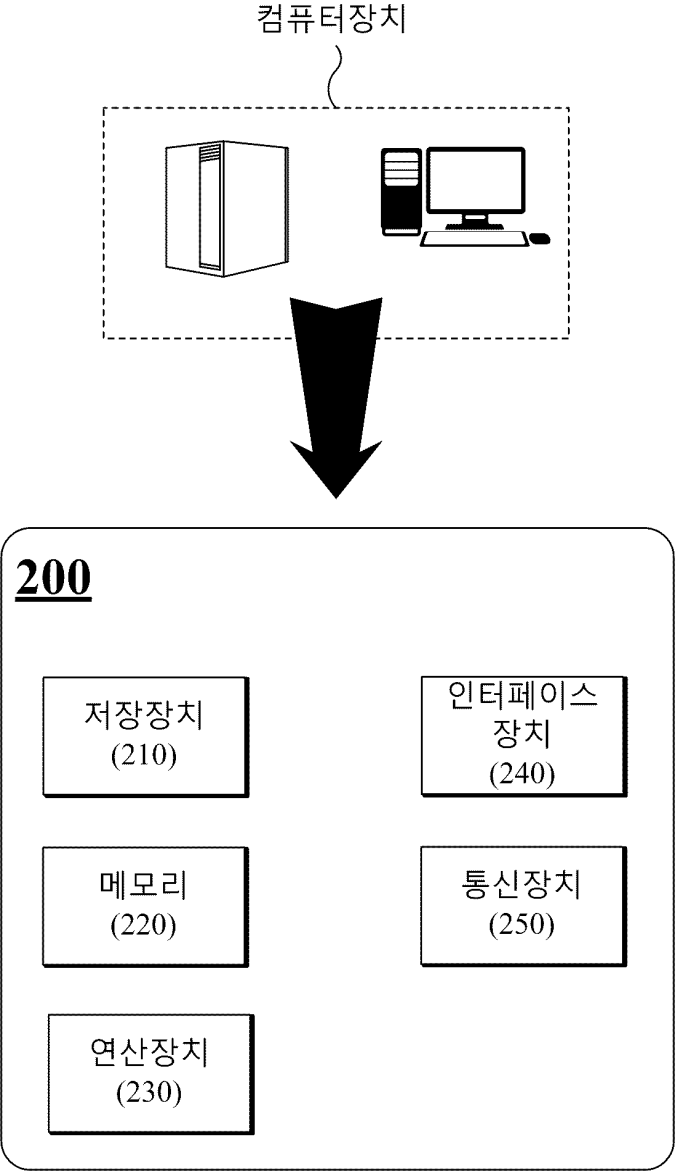
【도 4】



【도 5】



【도 6】



This is a preprint article, it offers immediate access but has not been peer reviewed.


[Download This Paper](#)
[Open PDF in Browser](#)

Share:


[Add Paper to My Library](#)

# K2vtune: Automatic Database Tuning with Knob Vector Representation

18 Pages • Posted: 21 Sep 2022

[Jieun Lee](#)

Yonsei University

[Jonghwan Choi](#)

Yonsei University

[Sangmin Seo](#)

Yonsei University

[Sanghyun Park](#)

Yonsei University

## Abstract

Database management system (DBMS) performance optimization has been studied extensively. Database knob tuning is performance optimization method that controls knobs, which are important elements that determine database performance. DBMSs have numerous knobs, which makes it difficult for users to determine appropriate knob values to enhance the performance of the DBMS. Even for database experts, determining these values is a time-consuming task. Thus, an approach of automatically tuning the database knobs is required. To improve performance with automatic knob tuning, we propose a database knob tuning system, K2vTune, with a deep neural network. K2vTune first finds a workload that resembles the target workload. We then train a predictive model to be employed as a fitness function in genetic algorithm. For the predictive model, we propose a novel method Knob2vec to generate knob feature vectors by learning workload characteristics. We employ recurrent neural network-based models and attention mechanisms to build the predictive model. Through this model structure, the correlation between knobs and database performance is learned. The genetic algorithm recommends the knobs that can improve the database performance. We evaluate the performance of K2vTune in six different workloads via the RocksDB benchmark. The experimental results show that K2vTune yields significantly better configurations than the default configurations and outperforms existing knob tuning systems.

**Keywords:** Database Knob Tuning, Attention, Genetic algorithm

**Suggested Citation:**

Do you have a job opening that you'd like to promote on SSRN?

[Place Job Opening](#)

## Paper statistics

DOWNLOADS

8

ABSTRACT VIEWS

22

PlumX Metrics



## Related eJournals

Web Technology eJournal

[Follow](#)

Information Management eJournal

[Follow](#)
[View more >](#)

This is a preprint article, it offers immediate access but has not been peer reviewed.


[Download This Paper](#)
[Open PDF in Browser](#)

Share:


[Add Paper to My Library](#)

# K2vtune: Automatic Database Tuning with Knob Vector Representation

18 Pages • Posted: 21 Sep 2022

[Jieun Lee](#)

Yonsei University

[Jonghwan Choi](#)

Yonsei University

[Sangmin Seo](#)

Yonsei University

[Sanghyun Park](#)

Yonsei University

## Abstract

Database management system (DBMS) performance optimization has been studied extensively. Database knob tuning is performance optimization method that controls knobs, which are important elements that determine database performance. DBMSs have numerous knobs, which makes it difficult for users to determine appropriate knob values to enhance the performance of the DBMS. Even for database experts, determining these values is a time-consuming task. Thus, an approach of automatically tuning the database knobs is required. To improve performance with automatic knob tuning, we propose a database knob tuning system, K2vTune, with a deep neural network. K2vTune first finds a workload that resembles the target workload. We then train a predictive model to be employed as a fitness function in genetic algorithm. For the predictive model, we propose a novel method Knob2vec to generate knob feature vectors by learning workload characteristics. We employ recurrent neural network-based models and attention mechanisms to build the predictive model. Through this model structure, the correlation between knobs and database performance is learned. The genetic algorithm recommends the knobs that can improve the database performance. We evaluate the performance of K2vTune in six different workloads via the RocksDB benchmark. The experimental results show that K2vTune yields significantly better configurations than the default configurations and outperforms existing knob tuning systems.

**Keywords:** Database Knob Tuning, Attention, Genetic algorithm

**Suggested Citation:**

Do you have a job opening that you'd like to promote on SSRN?

[Place Job Opening](#)

## Paper statistics

DOWNLOADS

8

ABSTRACT VIEWS

22

PlumX Metrics



## Related eJournals

Web Technology eJournal

[Follow](#)

Information Management eJournal

[Follow](#)
[View more >](#)