



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0138905
(43) 공개일자 2022년10월14일

(51) 국제특허분류(Int. Cl.)
G06F 16/2453 (2019.01) G06F 16/22 (2019.01)
G06F 8/41 (2018.01) G06N 20/00 (2019.01)
(52) CPC특허분류
G06F 16/24544 (2019.01)
G06F 16/2282 (2019.01)
(21) 출원번호 10-2021-0044542
(22) 출원일자 2021년04월06일
심사청구일자 없음

(71) 출원인
한국전력공사
전라남도 나주시 전력로 55(빛가람동)
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
이규철
대전광역시 유성구 배울2로 133 (용산동, 경남아
너스빌2단지아파트) 205동 203호
김인아
대전광역시 유성구 대학로81번길 32-36 (궁동, 레
츠빌)
이경민
대전광역시 서구 관저로 51, 512동 2004호 (관저
동, 구봉마을)
(74) 대리인
특허법인아주

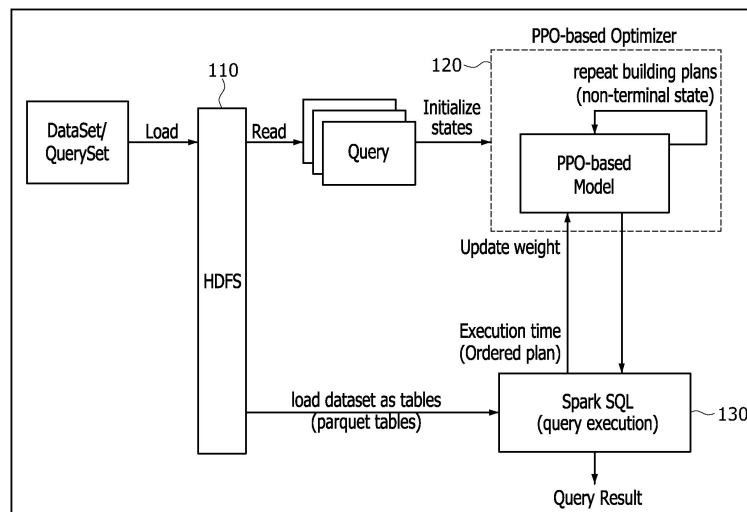
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 PPO 모델 기반 SPARK SQL 조인 최적화 장치 및 방법

(57) 요약

본 발명은 PPO 모델 기반 SPARK SQL 조인 최적화 장치에 관한 것으로, PPO(Proximal Policy Optimization) 모델을 학습시키기 위한 데이터 셋과 쿼리 셋(Dataset/QuerySet)을 로드하는 하둡 분산 파일 시스템(HDFS); 상기 하둡 분산 파일 시스템에서 리드한 쿼리를 입력받아 내부의 PPO 모델에 의해 특정 조인 순서를 가진 쿼리 플랜을 생성하는 PPO 기반 최적화부; 및 상기 HDFS에서 데이터 셋을 테이블로 읽어 상기 쿼리 플랜을 Spark SQL에 의해 실행 시간을 도출하는 Spark SQL부;를 포함하되, 상기 PPO 기반 최적화부는, 상기 쿼리 플랜의 실행 시간을 보상(reward)으로 사용하여 상기 PPO 모델을 학습한다.

대표도



(52) CPC특허분류

G06F 8/427 (2013.01)

G06N 20/00 (2021.08)

명세서

청구범위

청구항 1

PPO(Proximal Policy Optimization) 모델을 학습시키기 위한 데이터 셋과 쿼리 셋(Dataset/QuerySet)을 로드하는 하둡 분산 파일 시스템(HDFS);

상기 하둡 분산 파일 시스템에서 리드한 쿼리를 입력받아 내부의 PPO 모델에 의해 특정 조인 순서를 가진 쿼리 플랜을 생성하는 PPO 기반 최적화부; 및

상기 HDFS에서 데이터 셋을 테이블로 읽어 상기 쿼리 플랜을 Spark SQL에 의해 실행 시간을 도출하는 Spark SQL 부;를 포함하되,

상기 PPO 기반 최적화부는, 상기 쿼리 플랜의 실행 시간을 보상(reward)으로 사용하여 상기 PPO 모델을 학습하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 장치.

청구항 2

제 1항에 있어서, 상기 PPO 기반 최적화부는,

모델 학습이 시작되면, 쿼리(Query)를 상기 하둡 분산 파일 시스템에서 읽어와 PPO 모델의 입력으로 사용하기 위해 쿼리를 파싱하고, 파싱된 정보를 기반으로 강화학습 환경(Environment)의 상태(State)와 행동(Action)을 초기화(Initialize)하는 초기화 작업을 수행하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 장치.

청구항 3

제 2항에 있어서, 상기 PPO 기반 최적화부는,

쿼리의 정보를, 릴레이션 상태, 조인 술어 상태, 및 셀렉션 상태인 3가지 상태(State)로 표현하고, 쿼리의 정보에 따라, 조인이 가능한 임의의 두 릴레이션을 조인하는 형식을 가진 행동에 해당하는 행동(action)의 기본 형식을 설정하고, 상기 3가지 상태와 행동 형식을 PPO 모델의 입력으로 사용하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 장치.

청구항 4

제 1항에 있어서, 상기 PPO 모델은,

입력된 상태를 기반으로 하여 가능한 행동들의 후보를 도출하고 행동을 선택하며, 환경에서는 선택된 행동을 기반으로 기존의 상태를 새로운 상태로 업데이트하고, 행동에 대한 보상을 에이전트에게 전달하되, 이 때 최종 조인 순서가 도출되지 않았으면 항상 0값으로 전달되며, 최종적으로 조인 순서가 도출되면 PPO 기반 최적화부와 Spark SQL에 의해 최종 조인 순서에 대한 보상이 결정되고,

상기 에이전트는 새로운 상태를 기반으로 하여 행동을 선택하는 과정을 계속해서 반복하며, 더 이상 선택할 행동이 없이, 조인 대상 릴레이션의 수가 1이 되는 종료 상태가 될 때까지 이 과정을 반복하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 장치.

청구항 5

제 1항에 있어서, 상기 PPO 기반 최적화부는,

상기 PPO 모델이 종료 상태가 된 경우 도출된 조인 순서를 기반으로 Spark SQL이 인식할 수 있는 형태의 쿼리 플랜을 생성하여 Spark SQL부에 전달하며,

상기 Spark SQL부는 HDFS에 저장된 데이터 셋을 테이블로 읽어 이를 기반으로 상기 쿼리 플랜을 Spark SQL에서 실행하여 실제 실행 시간을 도출하여 이를 다시 상기 PPO 기반 최적화부에 전달하며,

상기 PPO 기반 최적화부에 지정된 보상 함수에 의해 보상으로 변환되어, 상기 PPO 모델을 학습하는데 사용하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 장치.

청구항 6

제 1항에 있어서, 상기 PPO 기반 최적화부는,

도출한 쿼리 플랜의 실행 시간을, 실행 시간($Time_n$)에 역수를 취하고, 실행 시간이 많이 소요되는 플랜에 대해 임계 시간(k)을 정하여, 임계 시간을 초과할 경우 보상을 -1로 부여하여 이를 기준으로 실행 시간을 보상(Reward)으로 변환하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 장치.

청구항 7

하둡 분산 파일 시스템(HDFS)이 PPO(Proximal Policy Optimization) 모델을 학습시키기 위한 데이터 셋과 쿼리 셋(Dataset/QuerySet)을 로드하는 단계;

PPO 기반 최적화부가 상기 하둡 분산 파일 시스템에서 리드한 쿼리를 입력받아 내부의 PPO 모델에 의해 특정 조인 순서를 가진 쿼리 플랜을 생성하는 단계;

Spark SQL부가 HDFS에서 데이터 셋을 테이블로 읽어 상기 쿼리 플랜을 Spark SQL에 의해 실행하여 실행 시간을 도출하는 단계; 및

상기 PPO 기반 최적화부가 실행 시간을 보상(reward)으로 사용하여 상기 PPO 모델을 학습하는 단계;를 포함하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 방법.

청구항 8

제 7항에 있어서, 상기 모델 학습이 시작되면,

상기 PPO 기반 최적화부는,

쿼리(Query)를 상기 하둡 분산 파일 시스템에서 읽어와 PPO 모델의 입력으로 사용하기 위해 쿼리를 파싱하고, 파싱된 정보를 기반으로 강화학습 환경(Environment)의 상태(State)와 행동(Action)을 초기화(Initialize)하는 초기화 작업을 수행하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 방법.

청구항 9

제 8항에 있어서, 상기 PPO 기반 최적화부는,

쿼리의 정보를, 릴레이션 상태, 조인 술어 상태, 및 선택성 상태인 3가지 상태(State)로 표현하고, 쿼리의 정보에 따라, 조인이 가능한 임의의 두 릴레이션을 조인하는 형식을 가진 행동에 해당하는 행동(action)의 기본 형식을 설정하고, 상기 3가지 상태와 행동 형식을 PPO 모델의 입력으로 사용하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 방법.

청구항 10

제 7항에 있어서, 상기 PPO 모델은,

입력된 상태를 기반으로 하여 가능한 행동들의 후보를 도출하고 행동을 선택하며, 선택된 행동을 기반으로 기존의 상태를 새로운 상태로 업데이트하고, 행동에 대한 보상을 에이전트에게 전달하되, 이 때 최종 조인 순서가 도출되지 않았으면 항상 0값으로 전달되며, 최종적으로 조인 순서가 도출되면 PPO 기반 최적화부와 Spark SQL에 의해 최종 조인 순서에 대한 보상이 결정되고,

상기 에이전트는 새로운 상태를 기반으로 하여 행동을 선택하는 과정을 계속해서 반복하며, 더 이상 선택할 행동이 없이, 조인 대상 릴레이션의 수가 1이 되는 종료 상태가 될 때까지 이 과정을 반복하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 방법.

청구항 11

제 7항에 있어서, 상기 PPO 모델이 종료 상태가 된 경우,

상기 PPO 기반 최적화부는, 도출된 조인 순서를 기반으로 Spark SQL이 인식할 수 있는 형태의 쿼리 플랜을 생성하여 Spark SQL부에 전달하며,

상기 Spark SQL부는 HDFS에 저장된 데이터 셋을 테이블로 읽어 이를 기반으로 상기 쿼리 플랜을 Spark SQL에서 실행하여 실제 실행 시간을 도출하여 이를 다시 상기 PPO 기반 최적화부에 전달하며,

상기 PPO 기반 최적부에 지정된 보상 함수에 의해 보상으로 변환되어, 상기 PPO 모델을 학습하는데 사용하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 방법.

청구항 12

제 7항에 있어서, 상기 PPO 기반 최적화부는,

도출한 쿼리 플랜의 실행 시간을, 실행 시간($Time_n$)에 역수를 취하고, 실행 시간이 많이 소요되는 플랜에 대해 임계 시간(k)을 정하여, 임계 시간을 초과할 경우 보상을 -1로 부여하여 이를 기준으로 실행 시간을 보상(Reward)으로 변환하는 것을 특징으로 하는 PPO 모델 기반 SPARK SQL 조인 최적화 방법.

발명의 설명

기술 분야

[0001] 본 발명은 PPO 모델 기반 SPARK SQL 조인 최적화 장치 및 방법에 관한 것으로, 보다 상세하게는 강화학습의 알고리즘인 PPO(Proximal Policy Optimization) 알고리즘을 사용하여 실제 쿼리(query)의 실행 시간을 학습한 모델을 생성하고, 이를 SPARK SQL과 결합한 PPO 모델 기반 SPARK SQL 조인 최적화 장치 및 방법에 관한 것이다.

배경 기술

[0002] 일반적으로 계량 데이터 관리 시스템(MDMS : Meter Data Management System)은 지능형 전력 계량 인프라(AMI : Advanced Metering Infrastructure)로부터 발생한 AMI 계량 데이터를 분산 및 병렬 처리하여 실시간으로 관리하는 통합 시스템이다. 이러한 지능형 전력 계량 인프라에서는 전기 생산 시설, 변전소, 및 건물 내부의 사물인터넷으로부터 다양한 데이터들이 발생하며, 연간 수천 페타바이트의 대용량 데이터가 생성된다.

[0003] 이 때 상기 생성되는 데이터 유형은 아래의 표 1과 같이 크게 3가지로 구분되며, 예컨대 전력 생산 데이터, 전력 송·배전 데이터, 계기 마스터 정보, 및 전력 검침 계량 데이터 등이 포함된다.

표 1

[0004]

종류	특징
미터 데이터	AMI 스마트 미터로부터 수집된 데이터
아카이브 데이터	미터 데이터에 대한 데이터 (예 : 사용자, 디바이스 등에 대한 정보)
통계 데이터	미터/아카이브 데이터 기반 온라인 배치 분석 결과

- [0005] 이 때 상기 대량의 계량 데이터들은 실시간으로 분산 및 병렬 처리를 수행하는 계량 데이터 관리 시스템에서 사용되며, 상기 계량 데이터 관리 시스템(MDMS)은 수집, 검증, 집계, 및 분석하는 작업을 수행하거나 설비 원격제어를 수행한다.
- [0006] 상기 계량 데이터 관리 시스템(MDMS)은 데이터를 집계 및 분석 하거나, 전기 사용량 조회 등의 작업을 수행할 때, 쿼리 요청이 지속적으로 발생하며, 전력 관련 데이터를 실시간으로 관리해야하는 시스템이기 때문에 쿼리의 신속한 처리가 필요하다. 하지만 상기 계량 데이터 관리 시스템(MDMS)에서 발생하는 쿼리는 많은 종류의 입력 테이블과 조인, 선택, 및 집계 연산 등의 다양한 연산이 포함되며, 특히 그 중에서도 연산 비용이 높은 조인 연산이 많이 발생하기 때문에 높은 연산 복잡도를 가진다.
- [0007] 따라서 연산의 복잡도를 낮추고 쿼리 실행 시간을 최소화하기 위한 쿼리 최적화(Query optimization) 과정이 요구된다.
- [0008] 상기 쿼리 최적화(Query optimization)는 입력 쿼리에 대해 가능한 플랜을 고려하여 쿼리를 실행하는 가장 효율적인 순서와 방법을 결정하는 것이다.
- [0009] 참고로 기존 쿼리 최적화 방법은 입력 쿼리에 대해 가능한 연산과 방법에 따라 쿼리 플랜을 생성(Plan space enumeration)하고, 입력 테이블들의 통계 정보(Cardinality estimation)를 기반으로 생성한 쿼리 플랜들의 비용(Cost)들을 비교하여 가장 낮은 비용을 가지는 쿼리 플랜(Optimal query plan)을 선택하여 실행하는 방법을 사용하는데 이러한 방법은 아래의 세 가지 문제점이 있다.
- [0010] 첫 번째 문제점은 휴리스틱(Heuristic)한 최적화 알고리즘으로 인한 낮은 검색 범위(Search Space)를 갖는 것이다. 즉, 기존 쿼리 최적화 알고리즘은 최적화 수행 복잡도와 쿼리 실행 시간 사이의 트레이드오프(Trade-off)를 고려하여, 가능한 모든 조인 순서의 경우를 고려하지 않고 일부의 경우만을 고려하는 휴리스틱(Heuristic)한 방법을 사용한다. 그러나 이런 방법으로 인해 기존 최적화 방식은 최적의 조인 순서 플랜을 탐색하지 못할 가능성이 있다.
- [0011] 두 번째 문제점은 연산이 증가함에 따른 전체 비용 예측 정확도가 감소된다는 것이다. 즉, 기존 쿼리 최적화 알고리즘은 입력 쿼리의 테이블로부터 추정된 통계 정보를 사용하여 쿼리 플랜의 비용을 계산한다. 이에 연산이 복잡해지고, 증가할수록 비용 예측의 기준이 되는 중간 결과 크기의 예측이 부정확해지고 전체 비용 예측 정확도는 더 감소하는 문제점이 발생한다.
- [0012] 세 번째 문제점은 과거에 쿼리 실행의 경험을 학습하지 못하는 정적인 최적화라는 것이다. 즉, 기존 쿼리 최적화 알고리즘은 정적인 알고리즘을 사용하여 최적화를 수행하며, 추정된 쿼리 플랜의 비용과 실제 쿼리 실행 시간의 차이가 클 수 있으나, 이러한 실행 경험을 학습하지 못하고 계속해서 반복되는 문제점이 있다.
- [0013] 따라서 상기와 같은 문제점을 고려하여 데이터베이스 최적화 기술에 강화 학습(Reinforcement Learning)을 적용하려는 시도가 증가하고 있으며, 최근 강화 학습을 사용하여 쿼리 최적화 문제를 개선하려는 연구들이 제안되었다.
- [0014] 상기 강화 학습은 풀고자 하는 문제를 환경(Environment)으로 표현하고, 행동(Action)을 취하는 에이전트(Agent)와 환경이 상호 작용하는 것으로 가정한다. 여기서 상기 환경은 에이전트에게 현재 상태(State)와 가능한 행동들의 집합을 알려주고, 상기 에이전트가 행동을 선택하면 상기 환경은 상기 에이전트에게 행동에 따른 보상(Reward)과 새로운 상태, 가능한 행동들의 집합을 제공하는 과정을 반복하고, 보상을 최대화하려는 행동들을 선택하도록 학습된다.
- [0015] 이 때 쿼리 최적화는, 도 1에 도시된 바와 같이, 입력 쿼리에 존재하는 조인 연산을 기반으로 조인할 대상 테이블 또는 서브쿼리를 선택하는 과정을 반복하여 상향식(Bottom-up)으로 쿼리 플랜을 생성하여 트리 형태로 구축하는 방식이다. 따라서 쿼리 최적화 문제는 강화 학습이 풀고자 하는 문제인 순차적 의사결정(Sequential decision making)과 유사한 문제로 볼 수 있다.
- [0016] 최근 강화학습 기반의 쿼리 최적화 연구들은 단일 환경 DBMS을 기반으로 하고 있다. 하지만, 상기 계량 데이터 관리 시스템(MDMS)은 대용량 데이터를 처리하는 시스템이기 때문에 단일 환경에서의 처리는 한계가 있으며, 분산 환경에서의 쿼리 최적화가 요구된다. 이에 대용량 데이터를 처리하기 위해 다양한 종류의 분산 처리 환경이 사용되고 있지만, 인메모리(In-memory) 방식의 Apache Spark가 주로 사용된다.
- [0017] 참고로 상기 Spark는 Spark core를 기반으로 쿼리를 처리하는 Spark SQL을 지원하며, 쿼리 최적화 및 실행 기능을 포함한다. 그런데 상기 Spark SQL 또한 휴리스틱하고 정적인 최적화 알고리즘을 사용하여, 상술한 세 가지

문제점을 포함한다.

[0018] 따라서 상기 기존의 문제점을 개선하기 위해서, 강화학습의 알고리즘인 PPO(Proximal Policy Optimization)를 사용하여 실제 쿼리의 실행 시간을 학습한 모델을 개발하고, 이를 Spark SQL과 결합한 조인 최적화 장치(또는 조인 최적화 시스템)의 개발이 필요한 상황이다. 여기서 상기 학습 모델은 Spark SQL의 기존 조인 최적화의 실행을 생략하고, 정적으로 정의된 알고리즘을 사용하여 동작하는 Spark SQL의 문제점을 고려하여, 실제 과거 쿼리 실행 경험을 학습하여 보다 빠른 쿼리 플랜을 도출할 수 있도록 한다.

[0019] 본 발명의 배경기술은 대한민국 등록특허 10-2127272호(2020.06.22. 등록, 통계적 SQL 패턴 분석을 통한 SQL 튜닝 자동화 방법 및 시스템)에 개시되어 있다.

발명의 내용

해결하려는 과제

[0020] 본 발명의 일 측면에 따르면, 본 발명은 상기와 같은 문제점을 해결하기 위해 창작된 것으로서, 강화학습의 알고리즘인 PPO(Proximal Policy Optimization) 알고리즘을 사용하여 실제 쿼리(query)의 실행 시간을 학습한 모델을 생성하고, 이를 SPARK SQL과 결합한 PPO 모델 기반 SPARK SQL 조인 최적화 장치 및 방법을 제공하는 데 그 목적이 있다.

과제의 해결 수단

[0021] 본 발명의 일 측면에 따른 PPO 모델 기반 SPARK SQL 조인 최적화 장치는, PPO(Proximal Policy Optimization) 모델을 학습시키기 위한 데이터 셋과 쿼리 셋(Dataset/QuerySet)을 로드하는 하둡 분산 파일 시스템(HDFS); 상기 하둡 분산 파일 시스템에서 리드한 쿼리를 입력받아 내부의 PPO 모델에 의해 특정 조인 순서를 가진 쿼리 플랜을 생성하는 PPO 기반 최적화부; 및 상기 HDFS에서 데이터 셋을 테이블로 읽어 상기 쿼리 플랜을 Spark SQL에 의해 실행 시간을 도출하는 Spark SQL부;를 포함하되, 상기 PPO 기반 최적화부는, 상기 쿼리 플랜의 실행 시간을 보상(reward)으로 사용하여 상기 PPO 모델을 학습하는 것을 특징으로 한다.

[0022] 본 발명에 있어서, 상기 PPO 기반 최적화부는, 모델 학습이 시작되면, 쿼리(Query)를 상기 하둡 분산 파일 시스템에서 읽어와 PPO 모델의 입력으로 사용하기 위해 쿼리를 파싱하고, 파싱된 정보를 기반으로 강화학습 환경(Environment)의 상태(State)와 행동(Action)을 초기화(Initialize)하는 초기화 작업을 수행하는 것을 특징으로 한다.

[0023] 본 발명에 있어서, 상기 PPO 기반 최적화부는, 쿼리의 정보를, 릴레이션 상태, 조인 술어 상태, 및 셀렉션 상태인 3가지 상태(State)로 표현하고, 쿼리의 정보에 따라, 조인이 가능한 임의의 두 릴레이션을 조인하는 형식을 가진 행동에 해당하는 행동(action)의 기본 형식을 설정하고, 상기 3가지 상태와 행동 형식을 PPO 모델의 입력으로 사용하는 것을 특징으로 한다.

[0024] 본 발명에 있어서, 상기 PPO 모델은, 입력된 상태를 기반으로 하여 가능한 행동들의 후보를 도출하고 행동을 선택하며, 환경에서는 선택된 행동을 기반으로 기존의 상태를 새로운 상태로 업데이트하고, 행동에 대한 보상에 에이전트에게 전달하되, 이 때 최종 조인 순서가 도출되지 않았으면 항상 0값으로 전달되며, 최종적으로 조인 순서가 도출되면 PPO 기반 최적화부와 Spark SQL에 의해 최종 조인 순서에 대한 보상이 결정되고, 상기 에이전트는 새로운 상태를 기반으로 하여 행동을 선택하는 과정을 계속해서 반복하며, 더 이상 선택할 행동이 없이, 조인 대상 릴레이션의 수가 1이 되는 종료 상태가 될 때까지 이 과정을 반복하는 것을 특징으로 한다.

[0025] 본 발명에 있어서, 상기 PPO 기반 최적화부는, 상기 PPO 모델이 종료 상태가 된 경우 도출된 조인 순서를 기반으로 Spark SQL이 인식할 수 있는 형태의 쿼리 플랜을 생성하여 Spark SQL부에 전달하며, 상기 Spark SQL부는 HDFS에 저장된 데이터 셋을 테이블로 읽어 이를 기반으로 상기 쿼리 플랜을 Spark SQL에서 실행하여 실제 실행 시간을 도출하여 이를 다시 상기 PPO 기반 최적화부에 전달하며, 상기 PPO 기반 최적화부에 지정된 보상 함수에 의해 보상으로 변환되어, 상기 PPO 모델을 학습하는데 사용하는 것을 특징으로 한다.

[0026] 본 발명에 있어서, 상기 PPO 기반 최적화부는, 도출한 쿼리 플랜의 실행 시간을, 실행 시간($Time_n$)에 역수를 취하고, 실행 시간이 많이 소요되는 플랜에 대해 임계 시간(k)을 정하여, 임계 시간을 초과할 경우 보상을 -1로 부여하여 이를 기준으로 실행 시간을 보상(Reward)으로 변환하는 것을 특징으로 한다.

[0027] 본 발명의 다른 측면에 따른 PPO 모델 기반 SPARK SQL 조인 최적화 방법은, 하둡 분산 파일 시스템(HDFS)이

PPO(Proximal Policy Optimization) 모델을 학습시키기 위한 데이터 셋과 쿼리 셋(Dataset/QuerySet)을 로드하는 단계; PPO 기반 최적화부가 상기 하둡 분산 파일 시스템에서 리드한 쿼리를 입력받아 내부의 PPO 모델에 의해 특정 조인 순서를 가진 쿼리 플랜을 생성하는 단계; Spark SQL부가 HDFS에서 데이터 셋을 테이블로 읽어 상기 쿼리 플랜을 Spark SQL에 의해 실행하여 실행 시간을 도출하는 단계; 및 상기 PPO 기반 최적화부가 실행 시간을 보상(reward)으로 사용하여 상기 PPO 모델을 학습하는 단계;를 포함하는 것을 특징으로 한다.

[0028] 본 발명에 있어서, 상기 모델 학습이 시작되면, 상기 PPO 기반 최적화부는, 쿼리(Query)를 상기 하둡 분산 파일 시스템에서 읽어와 PPO 모델의 입력으로 사용하기 위해 쿼리를 파싱하고, 파싱된 정보를 기반으로 강화학습 환경(Environment)의 상태(State)와 행동(Action)을 초기화(Initialize)하는 초기화 작업을 수행하는 것을 특징으로 한다.

[0029] 본 발명에 있어서, 상기 PPO 기반 최적화부는, 쿼리의 정보를, 릴레이션 상태, 조인 술어 상태, 및 선택션 상태인 3가지 상태(State)로 표현하고, 쿼리의 정보에 따라, 조인이 가능한 임의의 두 릴레이션을 조인하는 형식을 가진 행동에 해당하는 행동(action)의 기본 형식을 설정하고, 상기 3가지 상태와 행동 형식을 PPO 모델의 입력으로 사용하는 것을 특징으로 한다.

[0030] 본 발명에 있어서, 상기 PPO 모델은, 입력된 상태를 기반으로 하여 가능한 행동들의 후보를 도출하고 행동을 선택하며, 선택된 행동을 기반으로 기존의 상태를 새로운 상태로 업데이트하고, 행동에 대한 보상을 에이전트에게 전달하되, 이 때 최종 조인 순서가 도출되지 않았으면 항상 0값으로 전달되며, 최종적으로 조인 순서가 도출되면 PPO 기반 최적화부와 Spark SQL에 의해 최종 조인 순서에 대한 보상이 결정되고, 상기 에이전트는 새로운 상태를 기반으로 하여 행동을 선택하는 과정을 계속해서 반복하며, 더 이상 선택할 행동이 없이, 조인 대상 릴레이션의 수가 1이 되는 종료 상태가 될 때까지 이 과정을 반복하는 것을 특징으로 한다.

[0031] 본 발명에 있어서, 상기 PPO 모델이 종료 상태가 된 경우, 상기 PPO 기반 최적화부는, 도출된 조인 순서를 기반으로 Spark SQL이 인식할 수 있는 형태의 쿼리 플랜을 생성하여 Spark SQL부에 전달하며, 상기 Spark SQL부는 HDFS에 저장된 데이터 셋을 테이블로 읽어 이를 기반으로 상기 쿼리 플랜을 Spark SQL에서 실행하여 실제 실행 시간을 도출하여 이를 다시 상기 PPO 기반 최적화부에 전달하며, 상기 PPO 기반 최적화부에 지정된 보상 함수에 의해 보상으로 변환되어, 상기 PPO 모델을 학습하는데 사용하는 것을 특징으로 한다.

[0032] 본 발명에 있어서, 상기 PPO 기반 최적화부는, 도출한 쿼리 플랜의 실행 시간을, 실행 시간($Time_n$)에 역수를 취하고, 실행 시간이 많이 소요되는 플랜에 대해 임계 시간(k)을 정하여, 임계 시간을 초과할 경우 보상을 -1로 부여하여 이를 기준으로 실행 시간을 보상(Reward)으로 변환하는 것을 특징으로 한다.

발명의 효과

[0033] 본 발명의 일 측면에 따르면, 본 발명은 강화학습의 알고리즘인 PPO 알고리즘을 사용하여 실제 쿼리(query)의 실행 시간을 학습한 모델을 생성하고, 이를 SPARK SQL과 결합한 PPO 모델 기반 SPARK SQL 조인 최적화를 수행함으로써, 실제 과거 쿼리 실행 경험을 학습하여 보다 빠른 쿼리 플랜을 도출한다.

도면의 간단한 설명

- [0034] 도 1은 기존의 상향식 쿼리 플랜 생성 방법을 보인 예시도.
- 도 2는 본 발명의 일 실시예에 따른 PPO 모델 기반 SPARK SQL 조인 최적화 장치의 개략적인 구성을 보인 예시도.
- 도 3은 상기 도 2에 있어서, 강화학습 동작을 설명하기 위한 예시도.
- 도 4는 상기 도 2에 있어서, PPO 기반 조인 최적화 동작을 설명하기 위한 예시도.
- 도 5는 상기 도 2에 있어서, 릴레이션 상태 표현 방식을 보인 예시도.
- 도 6은 상기 도 2에 있어서, 조인 술어 상태 표현 방식을 보인 예시도.
- 도 7은 상기 도 2에 있어서, 선택션 상태 표현 방식을 보인 예시도.
- 도 8은 상기 도 2에 있어서, 특정 릴레이션 상태에서 행동 표현 방식을 보인 예시도.
- 도 9는 데이터 10GB에 대해 기존 Spark SQL의 쿼리 실행 시간과 본 실시예에 따른 PPO 모델이 도출한 쿼리 플랜

의 실행 시간을 상대적으로 나타낸 그래프.

도 10은 데이터 100GB에 대해 기존 Spark SQL의 쿼리 실행 시간과 본 실시예에 따른 PPO 모델이 도출한 쿼리 플랜의 실행 시간을 상대적으로 나타낸 그래프.

도 11은 데이터 300GB에 대해 기존 Spark SQL의 쿼리 실행 시간과 본 실시예에 따른 PPO 모델이 도출한 쿼리 플랜의 실행 시간을 상대적으로 나타낸 그래프.

발명을 실시하기 위한 구체적인 내용

- [0035] 이하, 첨부된 도면을 참조하여 본 발명에 따른 PPO 모델 기반 SPARK SQL 조인 최적화 장치 및 방법의 일 실시예를 설명한다.
- [0036] 이 과정에서 도면에 도시된 선들의 두께나 구성요소의 크기 등은 설명의 명료성과 편의상 과장되게 도시되어 있을 수 있다. 또한, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례에 따라 달라질 수 있다. 그러므로 이러한 용어들에 대한 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.
- [0037] 도 2는 본 발명의 일 실시예에 따른 PPO 모델 기반 SPARK SQL 조인 최적화 장치의 개략적인 구성을 보인 예시도이다.
- [0038] 도 2에 도시된 바와 같이, 본 실시예에 따른 PPO 모델 기반 SPARK SQL 조인 최적화 장치는, 하둡 분산 파일 시스템(HDFS : Hadoop Distributed File System)(110), PPO 기반 최적화부(PPO-based Optimizer)(120), 및 Spark SQL부(130)를 포함한다.
- [0039] 도 2에 도시된 PPO 모델 기반 SPARK SQL 조인 최적화 장치는, 쿼리 실행 경험(실행 시간)을 기반으로 PPO 알고리즘을 사용한 Spark SQL 조인 최적화 장치의 동작과정을 나타내는 것으로서, 상기 PPO 기반 최적화부(120)에 쿼리(Query)가 입력되면, PPO 모델(즉, PPO 기반 강화학습 모델, PPO-based Model)에 의해 특정 조인 순서를 가진 쿼리 플랜이 생성된다.
- [0040] 상기 생성된 쿼리 플랜은 Spark SQL이 도출된 조인 순서를 인식할 수 있는 형식으로 Spark SQL에 의해 실행되며, 쿼리 플랜의 실행 시간을 얻을 수 있으며, 이를 보상(reward)으로 사용하여 PPO 모델을 학습하는데 사용된다.
- [0041] 도 2를 참조하면, 데이터 셋/쿼리 셋(Dataset/QuerySet)에서 데이터 셋은 상기 Spark SQL부(130)에 의해 쿼리 플랜을 실행할 때 조회되는 데이터 셋이고, 쿼리 셋은 상기 PPO 기반 최적화부 내부의 PPO 모델을 학습시키기 위한 쿼리 셋이며, PPO 모델(즉, PPO 기반 강화학습 모델, PPO-based Model)을 학습시키기 이전에 하둡(Hadoop)에서 제공하는 분산 저장소인 하둡 분산 파일 시스템(HDFS)(110)에 로드한다.
- [0042] 상기 PPO 기반 최적화부(120)는, 모델 학습이 시작되면, 쿼리(Query)를 하둡 분산 파일 시스템(HDFS)(110)에서 읽어오고, 강화 학습 모델의 입력으로 사용하기 위해 쿼리를 파싱하고, 파싱된 정보를 기반으로 강화학습 환경(Environment)의 상태(State)와 행동(Action)을 초기화(Initialize)하는 초기화 작업을 수행한다.
- [0043] 상기 상태 초기화가 완료되면, PPO 모델의 네트워크는 상기 PPO 기반 최적화부에 의해 초기화된 상태와 행동을 입력받아 초기화된다.
- [0044] 이 때 상기 PPO 기반 최적화부(120)는, 쿼리의 정보를, 릴레이션 상태, 조인 술어 상태, 및 선택션 상태인 3가지 상태(State)로 표현하고, 쿼리의 정보에 따라, 행동(action)의 기본 형식을 설정하고(조인이 가능한 임의의 두 릴레이션을 조인하는 형식을 가진 행동), 상기 3가지 상태와 행동 형식을 PPO 모델의 입력으로 사용한다.
- [0045] 도 3의 (a)에 도시된 바와 같이, PPO 모델은 입력된 상태를 기반으로 하여 가능한 행동들의 후보를 도출하고 행동을 선택한다(Take action).
- [0046] 그런 다음, 도 3의 (b)에 도시된 바와 같이, 환경에서는 선택된 행동을 기반으로 기존의 상태를 새로운 상태로 업데이트하고, 행동에 대한 보상을 에이전트에게 전달한다(이는 최종 조인 순서가 도출되지 않았으므로 항상 0 값으로 전달되며, 최종적으로 조인 순서가 도출되면 PPO 기반 최적화부와 Spark SQL에 의해 최종 조인 순서에 대한 보상이 결정됨).
- [0047] 이에 상기 에이전트는 새로운 상태를 기반으로 하여 행동을 선택하는 과정을 계속해서 반복하며, 더 이상 선택할 행동(조인)이 없을 때까지 상기 과정을 반복한다(종료 상태). 이 때 입력된 쿼리 내의 릴레이션 수가 1이 되

어 더 이상 선택할 수 있는 조인 연산이 없는 경우를 종료 상태로 정의한다.

- [0048] 상기 종료 상태가 된 경우 상기 PPO 기반 최적화부는 도출된 조인 순서에 기반하여 Spark SQL부가 인식할 수 있는 형태인 쿼리 플랜을 생성하며, 상기 쿼리 플랜을 Spark SQL부(130)에서 실행하여 실제 실행 시간을 도출한다. 상기 도출된 실행 시간은 상기 PPO 기반 최적화부(120) 내 지정된 보상 함수에 의해 보상으로 변환되어, PPO 기반 최적화부(120)에서 PPO 모델(즉, PPO 기반 강화학습 모델)을 학습하는데 사용된다.
- [0049] 도 4는 상태와 행동이 변하는 과정을 예를 들어 표현하고 있다.
- [0050] 본 실시예는 쿼리 최적화에 대한 정보로써 사용자의 입력 쿼리를 환경(Environment), 행동(즉, 행동 또는 Action)으로 표현한다.
- [0051] 상기 쿼리는 크게 SELECT, FROM, WHERE, GROUP BY, ORDER BY 등으로 구성되며, 본 실시예에서는 FROM 절의 릴레이션(Relation), WHERE 절의 조인 연산(Join Operation), 선택 연산(Selection Operation)을 강화학습 환경의 상태(State)로 표현한다.
- [0052] 상태 표현은 입력 쿼리의 릴레이션 정보, 조인 연산 정보, 선택 연산 정보를 PPO 모델이 인식할 수 있도록 벡터(Vector) 형태로 상태를 표현한다.
- [0053] 상기 릴레이션 정보는 릴레이션 상태(Relation State)로 정의했으며, 릴레이션 간 조인된 상태를 표시한다.
- [0054] 이 때 릴레이션 벡터는 $N * N$ 의 형태로 표현하며, 여기서 N 은 학습하고자 하는 데이터베이스 스키마 내에 존재하는 릴레이션의 수이다.
- [0055] 도 5의 (a)에 도시된 ‘초기 릴레이션 상태’는 조인이 하나도 일어나지 않은 상태를 표현하고, 도 5의 (b)에 도시된 ‘첫 번째 조인 후 릴레이션 상태’는 A 릴레이션과 C를 조인할 경우 릴레이션 상태가 업데이트 된 경우를 표시한다.
- [0056] 이 때 조인 연산 정보는 조인 술어 상태(Join Predicate State)로 정의하며, 입력 쿼리에 포함된 릴레이션 간 조인 조건(Condition)을 나타내며, 특정 상태에서 가능한 릴레이션 간 조인 연산의 후보를 도출하기 위해 사용되는 벡터로서, 릴레이션 상태와 마찬가지로 $N * N$ 의 형태로 표현한다.
- [0057] 도 6을 참조하면, 쿼리 내에 존재하는 연산들은 1로 표시, 존재하지 않는 경우 0으로 표시된다.
- [0058] 한편 선택 연산(Selection Operation)에 대한 정보는 선택 연산 상태(Selection State)로 정의하며, 이는 입력 쿼리 내에 존재하는 선택 연산을 나타낸다. 도 7에 도시된 바와 같이, 상기 선택 연산 상태는 $1 * M$ 형태로 표현하며, 여기서 M 은 학습하고자 하는 데이터베이스 스키마 내에 존재하는 어트리뷰트(Attribute)의 수이다. 상기 선택 연산 상태는 입력 쿼리에서 사용된 어트리뷰트의 경우 1로 표시, 사용되지 않은 어트리뷰트의 경우 0으로 표시한다.
- [0059] 본 실시예는 릴레이션 간 조인 연산을 행동(Action)으로 정의한다.
- [0060] 상기 행동이 의미하는 정확한 정의는, 특정 릴레이션 상태에서 취할 수 있는 행동으로써 릴레이션 또는 서브 트리(Sub tree) 간 조인이 결정된 것을 의미한다.
- [0061] 상기 특정 릴레이션 상태에서 취할 수 있는 행동들(가능한 조인 후보)을 구하기 위해 조인 술어 상태를 사용하였다. 도 8에 도시된 바와 같이, 상기 특정 릴레이션 상태에서 취할 수 있는 행동은 0개 이상이며, 그 중 하나의 행동을 선택할 경우 릴레이션 상태 벡터가 업데이트 된다.
- [0062] 상기 PPO 모델(즉, PPO 기반 강화학습 모델)에서 보상(Reward)은 모델을 학습시키기 위한 지표로써, 특정 릴레이션 상태에서 취한 행동에 대해 좋은 행동인지 좋지 못한 행동인지 판단하는 지표가 된다.
- [0063] 본 실시예에서는 입력 쿼리로부터 PPO 모델이 특정 조인 순서를 갖는 쿼리 플랜을 생성하고, Spark SQL부(130)에서 생성한 쿼리 플랜을 실제로 실행하여 실행 시간을 도출하며, 도출한 쿼리 플랜의 실행 시간을 보상으로 표현한다. 본 실시예는 실행 시간을 보상으로 표현하여 학습함으로써, PPO 모델은 실제 쿼리의 과거 실행 경험을 학습하여 보다 정확한 쿼리 플랜을 도출하는 근거가 된다.
- [0064] 본 실시예에 따른 실행 시간 기반 최적화 모델에서는 PPO 모델이 도출한 쿼리 플랜을 실행하여 실행 시간을 얻을 수 있고, 이를 보상 함수를 사용하여 보상으로 변환하여 사용한다. 또한 비용 기반 최적화 모델과 마찬가지로 쿼리 실행 시간이 길어질수록 보상은 줄어들고, 실행 시간이 짧아질수록 쿼리 플랜의 보상은 높아진다.

[0065] 또한 아래의 수학적식 1과 같이, 상기 실행 시간 기반 최적화 모델에서는 실행 시간의 스케일 차이가 크지 않기 때문에, 실행 시간($Time_n$)에 역수를 취하여 보상으로 사용하였다. 하지만 실행 시간이 많이 소요되는 플랜이 지속적으로 선택될 경우 학습 시간이 많이 소요되기 때문에, 쿼리별로 임계 시간(k)을 정하여, 임계 시간을 초과할 경우 보상을 -1로 부여하였다.

수학적식 1

$$\begin{cases} \text{reward} = 1 / \text{time}_n & (\text{Time}_n \leq k) \\ \text{reward} = -1 & (\text{Time}_n > k) \end{cases}$$

[0067] 한편 본 실시예에 따른 PPO 모델 기반 조인 최적화 장치와 기존의 Spark SQL이 도출하는 쿼리 플랜의 실행시간을 비교하는 성능 평가를 진행한 결과, 도 9 내지 도 11에 도시된 그래프와 같이, PPO 모델이 실행한 대부분의 쿼리에 대해 기존 Spark SQL보다 짧은 실행시간 또는 유사한 실행 시간을 가지는 것을 알 수 있다. 도 9는 데이터 10GB에 대해 기존 Spark SQL의 쿼리 실행 시간과 본 실시예에 따른 PPO 모델이 도출한 쿼리 플랜의 실행 시간을 상대적으로 나타낸 그래프이고, 도 10은 데이터 100GB에 대해 기존 Spark SQL의 쿼리 실행 시간과 본 실시예에 따른 PPO 모델이 도출한 쿼리 플랜의 실행 시간을 상대적으로 나타낸 그래프이며, 도 11은 데이터 300GB에 대해 기존 Spark SQL의 쿼리 실행 시간과 본 실시예에 따른 PPO 모델이 도출한 쿼리 플랜의 실행 시간을 상대적으로 나타낸 그래프이다.

[0068] 상술한 바와 같이 기존의 강화학습 기반의 쿼리 최적화 연구들은 모두 단일 환경 DBMS에 맞추어져 있으며, 또한 대량 데이터 검색을 위한 분산 DBMS 프레임워크로 많이 활용되는 Spark SQL은 휴리스틱하고 정적인 최적화 알고리즘을 사용하는 것을 고려하여 기존 DBMS의 쿼리 최적화 기술이 가지는 동일한 문제점이 있다.

[0069] 반면에 본 실시예는 강화학습의 한 알고리즘인 PPO 알고리즘을 사용하여 실제 쿼리의 실행 시간을 학습한 모델을 개발하고, 이를 Spark SQL과 결합한 조인 최적화 장치를 제공하는 것으로서, 이 때 학습 모델은 Spark SQL의 기존 조인 최적화의 실행을 생략하고, 정적으로 정의된 알고리즘을 사용하여 동작하는 Spark SQL의 문제점을 고려하여, 실제 과거 쿼리 실행 경험을 학습하여 보다 빠른 쿼리 플랜을 도출하는 효과가 있다.

[0070] 또한 본 실시예는 대량의 데이터에 대한 쿼리 처리 성능을 개선하기 위해, 강화학습 알고리즘을 사용하여 분산 프레임워크인 Spark SQL에서의 조인 최적화를 수행하는 시스템으로서, 계량 데이터 관리 시스템(MDMS)와 같이 조회 대상 데이터의 종류가 다양하고, 많은 조인 연산이 포함된 복잡한 쿼리가 발생하는 환경에서의 쿼리 로그를 학습함으로써 해당 환경에서 발생하는 쿼리 처리 성능을 개선시킬 수 있도록 하는 효과가 있다.

[0071] 이상으로 본 발명은 도면에 도시된 실시예를 참고로 하여 설명되었으나, 이는 예시적인 것에 불과하며, 당해 기술이 속하는 분야에서 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서 본 발명의 기술적 보호범위는 아래의 특허청구범위에 의해서 정하여져야 할 것이다. 또한 본 명세서에서 설명된 구현은, 예컨대, 방법 또는 프로세스, 장치, 소프트웨어 프로그램, 데이터 스트림 또는 신호로 구현될 수 있다. 단일 형태의 구현의 맥락에서만 논의(예컨대, 방법으로서만 논의)되었더라도, 논의된 특징의 구현은 또한 다른 형태(예컨대, 장치 또는 프로그램)로도 구현될 수 있다. 장치는 적절한 하드웨어, 소프트웨어 및 펌웨어 등으로 구현될 수 있다. 방법은, 예컨대, 컴퓨터, 마이크로프로세서, 집적 회로 또는 프로그래밍 가능한 로직 디바이스 등을 포함하는 프로세싱 디바이스를 일반적으로 지칭하는 프로세서 등과 같은 장치에서 구현될 수 있다. 프로세서는 또한 최종-사용자 사이에 정보의 통신을 용이하게 하는 컴퓨터, 셀 폰, 휴대용/개인용 정보 단말기(personal digital assistant: "PDA") 및 다른 디바이스 등과 같은 통신 디바이스를 포함한다.

부호의 설명

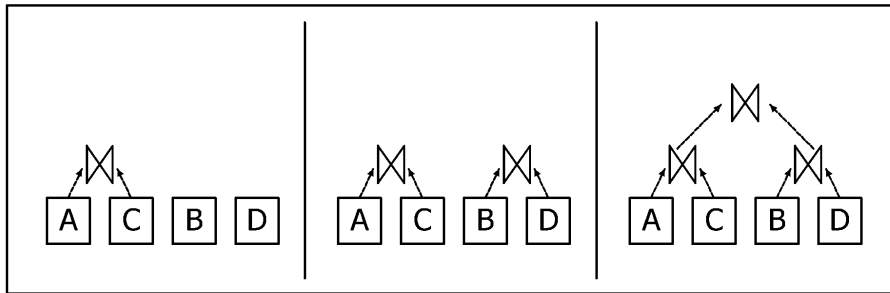
[0072] 110 : 하둡 분산 파일 시스템(HDFS)

120 : PPO 기반 최적화부

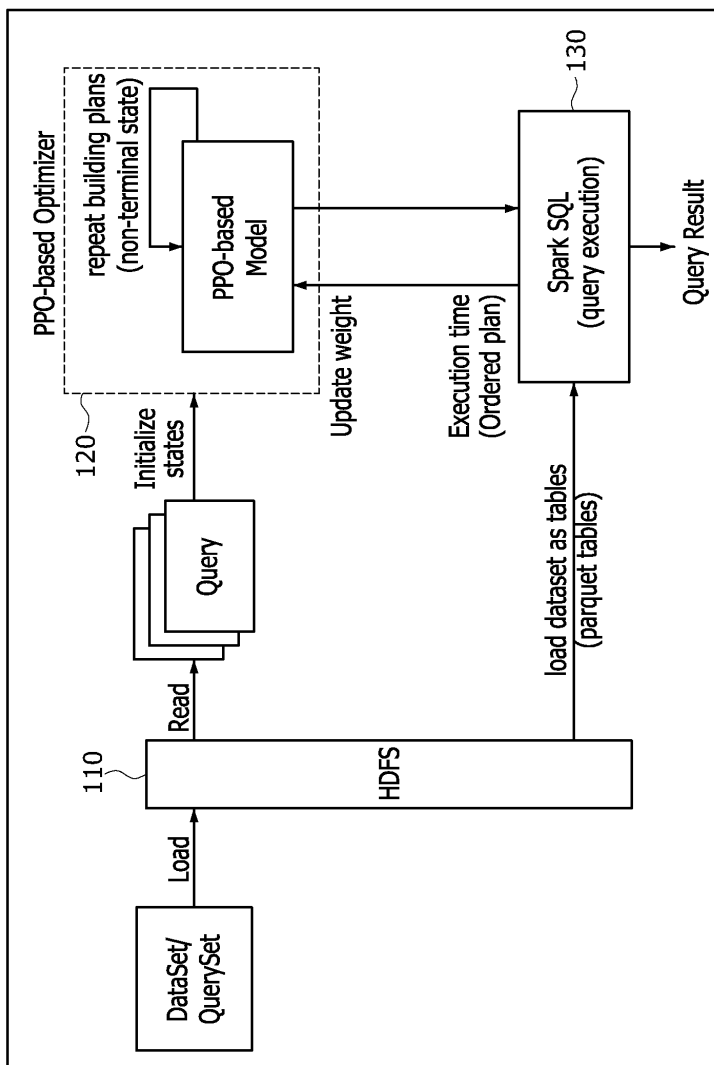
130 : Spark SQL부

도면

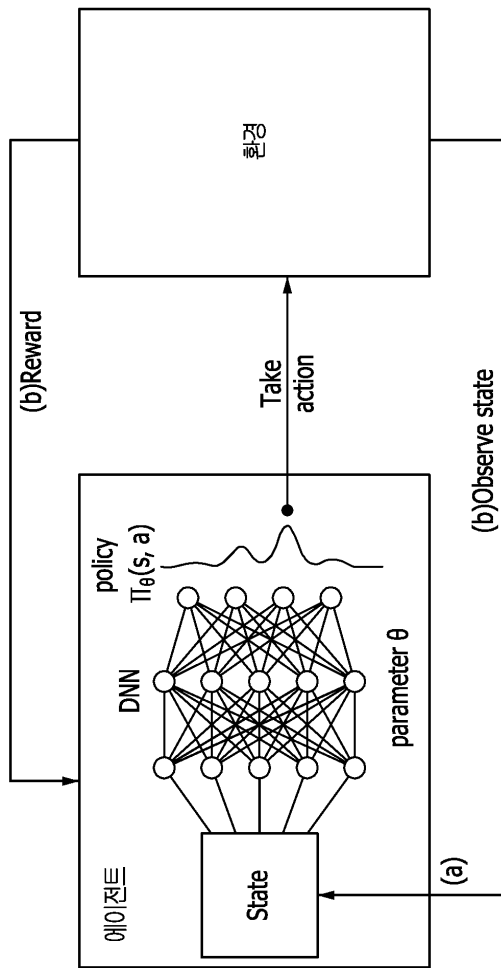
도면1



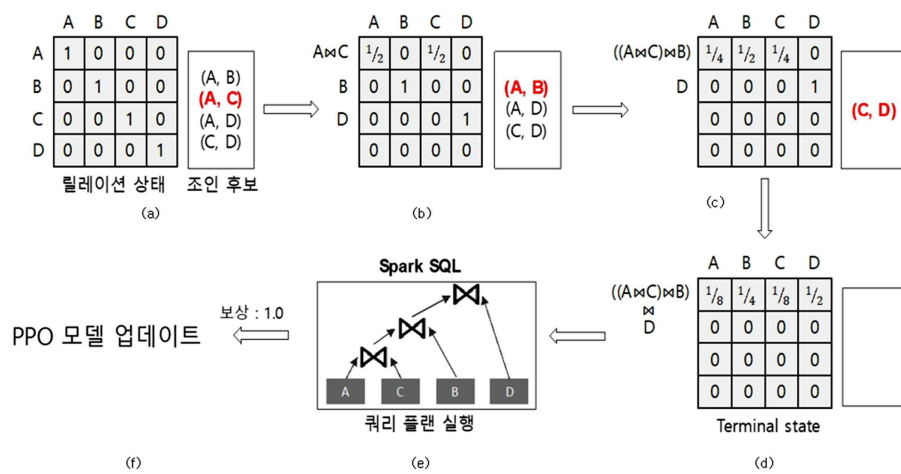
도면2



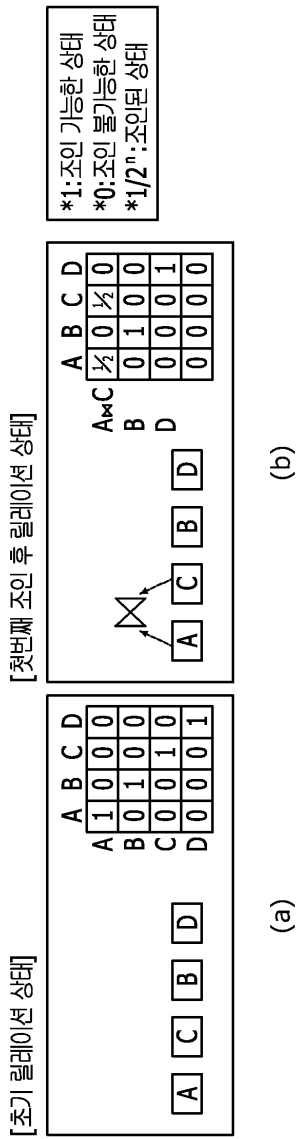
도면3



도면4



도면5



도면6

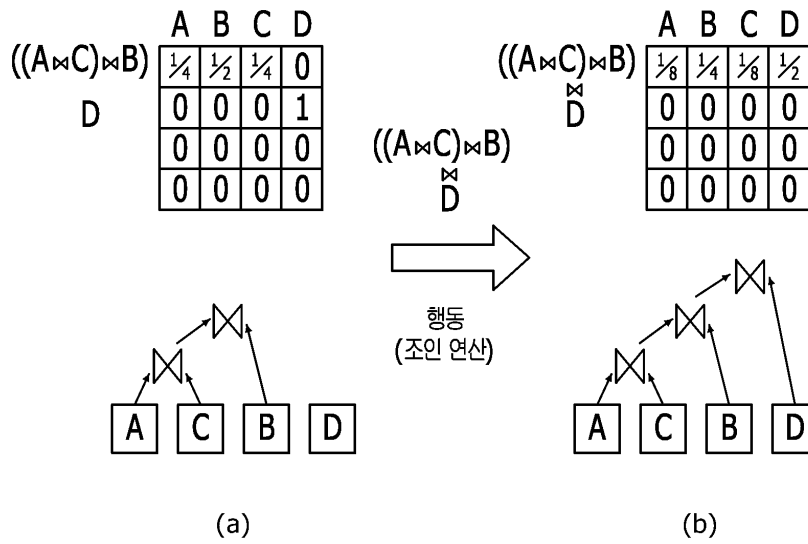
	A	B	C	D
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

*1 : 쿼리에 존재하는 조인
 *0 : 조인하지 않는 연산

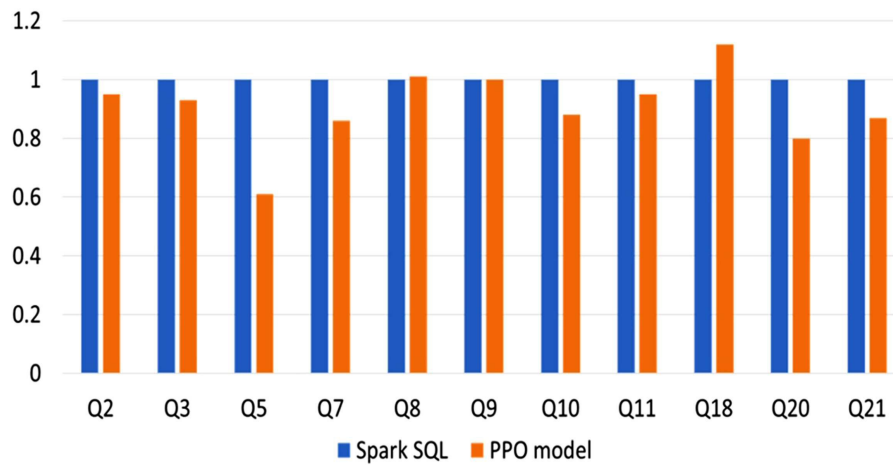
도면7

A.id	A.val	...	B.a2	C.id	...
0	0	...	1	0	0

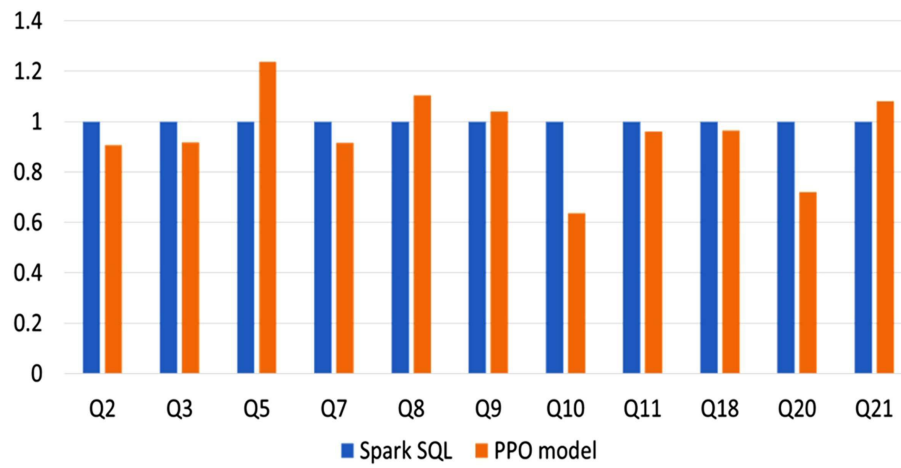
도면8



도면9



도면10



도면11

