



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2022년09월13일

(11) 등록번호 10-2442682

(24) 등록일자 2022년09월06일

(51) 국제특허분류(Int. Cl.)

G06N 99/00 (2019.01)

(52) CPC특허분류

G06N 20/00 (2021.08)

(21) 출원번호 10-2018-0120769

(22) 출원일자 2018년10월10일

심사청구일자 2021년10월12일

(65) 공개번호 10-2019-0094079

(43) 공개일자 2019년08월12일

(30) 우선권주장

62/625,532 2018년02월02일 미국(US)

15/942,218 2018년03월30일 미국(US)

(56) 선행기술조사문헌

US20180357234 A1

US20170169358 A1

US08095694 B

US20130271380 A1

(73) 특허권자

삼성전자주식회사

경기도 수원시 영통구 삼성로 129 (매탄동)

(72) 발명자

이주환

미국 캘리포니아주 95134 산호세 스카й탑 스트리트 52 217호

기양석

미국 캘리포니아주 94303 팔로알토 알테어 워크 873

(74) 대리인

특허법인 고려

전체 청구항 수 : 총 20 항

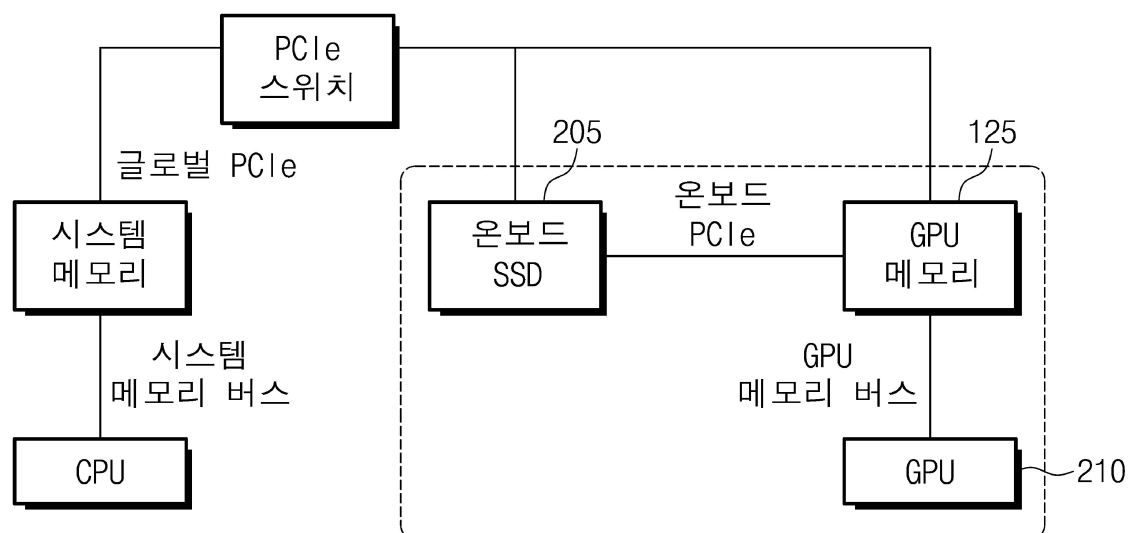
심사관 : 박승철

(54) 발명의 명칭 머신 러닝 시스템에서 직렬화된 키 값 접근을 방지하기 위한 시스템 및 방법

## (57) 요약

머신 러닝을 위한 시스템 및 방법에서, 시스템은 GPU 메모리를 갖는 GPU, 및 GPU 메모리에 연결된 키 값 스토리지 장치를 포함한다. 방법은 GPU에 의해, 키를 포함하는 키 값 요청을 GPU 메모리의 입력-출력 영역 내 키 값 요청 큐에 쓰는 단계를 포함한다. 방법은 키 값 스토리지 장치에 의해, 키 값 요청을 키 값 요청 큐로부터 읽는 단계, 및 키 값 스토리지 장치에 의해, 키 값 요청에 응답하여, 키 값 요청의 키에 대응하는 값을 GPU 메모리의 입력-출력 영역에 쓰는 단계를 더 포함한다.

## 대표도 - 도2



## 명세서

### 청구범위

#### 청구항 1

제 1 그래픽 프로세싱 유닛에 의해, 키를 포함하는 제 1 키 값 요청을 상기 제 1 그래픽 프로세싱 유닛에 연결된 제 1 메모리의 제 1 입력-출력 영역 내 키 값 요청 큐에 쓰는 단계;

상기 제 1 메모리에 연결된 제 1 키 값 스토리지 장치에 의해, 상기 제 1 입력-출력 영역에서 메모리가 매핑된 입력 및 출력 동작들을 수행하는 단계;

상기 제 1 키 값 스토리지 장치에 의해, 상기 제 1 키 값 요청을 상기 키 값 요청 큐로부터 읽는 단계; 및

상기 제 1 키 값 스토리지 장치에 의해, 상기 제 1 키 값 요청에 응답하여, 상기 제 1 키 값 요청의 상기 키에 대응하는 제 1 값을 상기 제 1 메모리의 상기 제 1 입력-출력 영역에 쓰는 단계를 포함하고,

상기 제 1 그래픽 프로세싱 유닛은 상기 제 1 메모리의 상기 제 1 입력-출력 영역에서 메모리가 매핑된 입력 및 출력 동작들을 수행하기 위해 상기 제 1 메모리에 메모리가 매핑되는 머신 러닝 시스템에서 키 값 접근을 위한 방법.

#### 청구항 2

제 1 항에 있어서,

상기 제 1 키 값 스토리지 장치에 의해, 상기 제 1 키 값 스토리지 장치에서, 키 검색을 수행하여 상기 제 1 값을 상기 제 1 메모리의 상기 입력-출력 영역에 쓰는 단계를 위해 상기 제 1 값을 회수하는 단계를 더 포함하는 방법.

#### 청구항 3

제 1 항에 있어서,

상기 제 1 키 값 요청은 상기 제 1 값에 할당된 영역인 리턴-값(return-value) 영역을 포함하는 방법.

#### 청구항 4

제 3 항에 있어서,

상기 제 1 값을 상기 제 1 메모리의 상기 제 1 입력-출력 영역에 쓰는 단계는 상기 제 1 값을 상기 리턴-값 영역에 쓰는 단계를 포함하는 방법.

#### 청구항 5

제 1 항에 있어서,

상기 제 1 값을 상기 제 1 메모리의 상기 제 1 입력-출력 영역에 쓰는 단계는 상기 제 1 값을 상기 제 1 메모리의 상기 제 1 입력-출력 영역 내 리턴-값 큐에 쓰는 단계를 포함하는 방법.

#### 청구항 6

제 1 항에 있어서,

상기 제 1 키 값 스토리지 장치 및 상기 제 1 그래픽 프로세싱 유닛에 연결된 호스트에 의해:

키 값 요청들을 수신하기 위해 상기 제 1 메모리의 상기 제 1 입력-출력 영역을 접근하고 그리고 상기 키 값 요청들에 응답하여 값들을 쓰는 상기 제 1 키 값 스토리지 장치; 그리고

상기 제 1 메모리의 상기 제 1 입력-출력 영역 내 키 값 요청들을 저장하고 그리고 상기 제 1 메모리의 상기 제 1 입력-출력 영역으로부터 값들을 읽는 상기 제 1 그래픽 프로세싱 유닛을 구성하는 단계를 더 포함하는 방법.

## 청구항 7

제 6 항에 있어서,

상기 호스트에 연결된 제 2 그래픽 프로세싱 유닛에 의해, 키를 포함하는 제 2 키 값 요청을 상기 제 2 그래픽 프로세싱 유닛에 연결된 제 2 메모리의 입력-출력 영역 내 키 값 요청 큐에 쓰는 단계;

상기 호스트 및 상기 제 2 메모리에 연결된 제 2 키 값 스토리지 장치에 의해, 상기 제 2 키 값 요청을 상기 키 값 요청 큐로부터 읽는 단계, 및

상기 제 2 키 값 스토리지 장치에 의해, 상기 제 2 키 값 요청에 응답하여, 상기 제 2 키 값 요청의 상기 키에 대응하는 제 2 값을 상기 제 2 메모리의 상기 입력-출력 영역에 쓰는 단계를 더 포함하는 방법.

## 청구항 8

제 7 항에 있어서,

상기 제 1 키 값 스토리지 장치에 의해, 상기 제 1 키 값 스토리지 장치에서, 키 검색을 수행하여 상기 제 1 값을 회수하는 단계; 및

상기 제 1 키 값 스토리지 장치에 의한 상기 키 검색을 수행하는 단계와 동시에, 상기 제 2 키 값 스토리지 장치에 의해, 상기 제 2 키 값 스토리지 장치에서, 키 검색을 수행하여 상기 제 2 값을 회수하는 단계를 더 포함하는 방법.

## 청구항 9

제 1 항에 있어서,

상기 제 1 키 값 스토리지 장치에 의해, 상기 제 1 키 값 요청을 읽는 단계는 P2P(peer-to-peer) DMA(direct memory access)를 통해 상기 제 1 키 값 요청을 읽는 단계를 포함하는 방법.

## 청구항 10

제 1 항에 있어서,

상기 제 1 키 값 스토리지 장치에 의해, 상기 제 1 값을 쓰는 단계는 P2P(peer-to-peer) DMA(direct memory access)를 통해 상기 제 1 값을 쓰는 단계를 포함하는 방법.

## 청구항 11

제 10 항에 있어서,

상기 제 1 키 값 스토리지 장치는 PCI(peripheral component interconnect) 연결에 의해 상기 제 1 그래픽 프로세싱 유닛에 연결되는 방법.

## 청구항 12

제 1 항에 있어서,

상기 제 1 그래픽 프로세싱 유닛에 의해, 상기 제 1 키 값 요청을 쓰는 단계 이후 그리고 상기 제 1 키 값 스토리지 장치에 의해, 상기 제 1 값을 상기 제 1 메모리의 상기 제 1 입력-출력 영역에 쓰는 단계 이전에, 상기 제 1 그래픽 프로세싱 유닛에 의해, 제 2 키 값 요청을 상기 키 값 요청 큐에 쓰는 단계를 더 포함하는 방법.

## 청구항 13

제 1 항에 있어서,

상기 제 1 그래픽 프로세싱 유닛에 의해, 키를 포함하는 제 2 키 값 요청을 상기 제 1 메모리의 제 2 입력-출력 영역 내 키 값 요청 큐에 쓰는 단계;

상기 제 1 메모리에 연결된 제 2 키 값 스토리지 장치에 의해, 상기 제 2 키 값 요청을 상기 제 1 메모리의 상기 제 2 입력-출력 영역의 상기 키 값 요청 큐로부터 읽는 단계, 및

상기 제 2 키 값 스토리지 장치에 의해, 상기 제 2 키 값 요청에 응답하여, 상기 제 2 키 값 요청의 상기 키에 대응하는 제 2 값을 상기 제 1 메모리의 상기 제 2 입력-출력 영역에 쓰는 단계를 더 포함하는 방법.

#### 청구항 14

제 13 항에 있어서,

상기 제 1 키 값 스토리지 장치에 의해, 상기 제 1 키 값 스토리지 장치에서, 키 검색을 수행하여 상기 제 1 값을 회수하는 단계, 및

상기 제 1 키 값 스토리지 장치에 의한 상기 키 검색을 수행하는 단계와 동시에, 상기 제 2 키 값 스토리지 장치에 의해, 상기 제 2 키 값 스토리지 장치에서, 키 검색을 수행하여 상기 제 2 값을 회수하는 단계를 더 포함하는 방법.

#### 청구항 15

그래픽 프로세싱 유닛;

상기 그래픽 프로세싱 유닛에 연결된 메모리; 및

키 값 스토리지 장치를 포함하되,

상기 키 값 스토리지 장치는 PCI(peripheral component interconnect) 연결에 의해 상기 그래픽 프로세싱 유닛에 연결되고;

상기 그래픽 프로세싱 유닛은 상기 메모리의 입력-출력 영역 내 메모리가 매핑(mapping)된 입력 및 출력 동작들을 수행하고, 그리고 하나 이상의 키 값 요청들을 상기 입력-출력 영역 내 키 값 요청 큐에 쓰도록 구성되고,

상기 키 값 스토리지 장치는:

상기 입력-출력 영역에서 메모리가 매핑된 입력 및 출력 동작들을 수행하고;

상기 하나 이상의 키 값 요청들을 상기 키 값 요청 큐로부터 읽고; 그리고

상기 하나 이상의 키 값 요청들의 키 값 요청에 응답하여, 상기 키 값 요청의 키에 대응하는 값을 상기 메모리의 상기 입력-출력 영역에 쓰도록 구성되는 머신 러닝(machine learning)을 위한 시스템.

#### 청구항 16

제 15 항에 있어서,

상기 키 값 요청은 상기 값에 할당된 영역인 리턴-값(return-value) 영역을 포함하는 머신 러닝을 위한 시스템.

#### 청구항 17

제 16 항에 있어서,

상기 값을 상기 메모리의 상기 입력-출력 영역에 쓰는 단계는 상기 값을 상기 리턴-값 영역에 쓰는 단계를 포함하는 머신 러닝을 위한 시스템.

#### 청구항 18

제 15 항에 있어서,

상기 값을 상기 메모리의 상기 입력-출력 영역에 쓰는 단계는 상기 값을 상기 메모리의 상기 입력-출력 영역 내 리턴-값 큐에 쓰는 단계를 포함하는 머신 러닝을 위한 시스템.

#### 청구항 19

그래픽 프로세싱 유닛;

키 값 스토리지 장치; 및

상기 그래픽 프로세싱 유닛과 상기 키 값 스토리지 장치 간의 통신을 위한 공유된 메모리를 포함하되;

상기 그래픽 프로세싱 유닛은 상기 공유된 메모리를 통해 하나 이상의 키 값 요청들을 상기 키 값 스토리지 장치로 전송하도록 구성되고,

상기 키 값 스토리지 장치는:

상기 공유된 메모리에서 메모리가 매핑된 입력 및 출력 동작들을 수행하고;

상기 하나 이상의 키 값 요청들을 수신하고; 그리고

상기 하나 이상의 키 값 요청들의 키 값 요청에 응답하여, 상기 키 값 요청의 키에 대응하는 값을 상기 공유된 메모리를 통해 상기 그래픽 프로세싱 유닛에 전송하도록 구성되고,

상기 그래픽 프로세싱 유닛은 상기 공유된 메모리에서 메모리가 매핑된 입력 및 출력 동작들을 수행하기 위해 상기 공유된 메모리에 메모리가 매핑되는 머신 러닝을 위한 시스템.

## 청구항 20

제 19 항에 있어서,

상기 공유된 메모리는 상기 그래픽 프로세싱 유닛에 연결된 메모리를 포함하고, 그리고

PCI(peripheral component interconnect) 연결을 통한 P2P(peer-to-peer) DMA(direct memory access)를 통해 상기 키 값 스토리지 장치에 의해 접근되도록 구성되는 머신 러닝을 위한 시스템.

## 발명의 설명

### 기술 분야

[0001] 본 발명에 따른 실시 예들의 하나 이상의 양상들은 머신 러닝에 관한 것으로, 좀 더 구체적으로 머신 러닝 시스템에서 직렬화된 키 값 접근을 방지하기 위한 시스템 및 방법에 관한 것이다.

### 배경 기술

[0002] 블록 인터페이스를 갖는 일부 종래 기술의 SSD(solid state drive)들에서, SSD에 저장된 데이터에 대한 키 값 접근(access)은 CPU(central processing unit)를 포함하여 전체 트레이닝 데이터의 부분 집합을 랜덤하게 (randomly) 샘플링하는 확률적(stochastic) 머신 러닝 동안 키 값 인터페이스를 제공하는 것을 요구한다. 호스트 CPU는 파일 인덱스 검색(lookup) 및 파일 시스템 접근을 수행하여 데이터의 위치를 식별하고, 이것은 직렬화된 키 값 접근으로 이어진다. 이러한 직렬화된 키 값 접근은 성능을 제한할 수 있다.

[0003] 따라서, 데이터에 대한 키 값 접근을 포함하는 머신 러닝을 수행하기 위한 개선된 시스템 및 방법이 필요하다.

## 발명의 내용

### 해결하려는 과제

[0004] 본 발명은 상술한 기술적 과제를 해결하기 위한 것으로, 본 발명은 머신 러닝 시스템에서 직렬화된 키 값 접근을 방지하기 위한 시스템 및 방법을 제공할 수 있다.

### 과제의 해결 수단

[0005] 본 발명의 실시 예에 따라 머신 러닝을 위한 방법이 제공되고, 방법은: 제 1 그래픽 프로세싱 유닛에 의해, 키를 포함하는 제 1 키 값 요청을 제 1 그래픽 프로세싱 유닛에 연결된 제 1 메모리의 제 1 입력-출력 영역 내 키 값 요청 큐에 쓰는 단계; 제 1 메모리에 연결된 제 1 키 값 스토리지 장치에 의해, 제 1 키 값 요청을 키 값 요청 큐로부터 읽는 단계, 및 제 1 키 값 스토리지 장치에 의해, 제 1 키 값 요청에 응답하여, 제 1 키 값 요청의 키에 대응하는 제 1 값을 제 1 메모리의 제 1 입력-출력 영역에 쓰는 단계를 포함한다.

[0006] 일 실시 예에 있어서, 방법은 제 1 키 값 스토리지 장치에 의해, 제 1 키 값 스토리지 장치에서, 키 검색을 수행하여 제 1 값을 회수하는 단계를 포함한다.

[0007] 일 실시 예에 있어서, 제 1 키 값 요청은 제 1 값에 할당된 영역인 리턴-값 영역을 포함한다.

- [0008] 일 실시 예에 있어서, 제 1 값을 제 1 메모리의 제 1 입력-출력 영역에 쓰는 단계는 제 1 값을 리턴-값 영역에 쓰는 단계를 포함한다.
- [0009] 일 실시 예에 있어서, 제 1 값을 제 1 메모리의 제 1 입력-출력 영역에 쓰는 단계는 제 1 값을 제 1 메모리의 제 1 입력-출력 영역 내 리턴-값 큐에 쓰는 단계를 포함한다.
- [0010] 일 실시 예에 있어서, 방법은 제 1 키 값 스토리지 장치 및 제 1 그래픽 프로세싱 유닛에 연결된 호스트에 의해: 제 1 키 값 스토리지 장치는 제 1 메모리의 제 1 입력-출력 영역을 접근하고, 키 값 요청들을 수신하고, 그리고 키 값 요청들에 응답하여 값들을 쓰고; 그리고 제 1 그래픽 프로세싱 유닛은 제 1 메모리의 제 1 입력-출력 영역 내 키 값 요청들을 저장하고 그리고 제 1 메모리의 제 1 입력-출력 영역으로부터 값들을 읽도록 구성하는 단계를 포함한다.
- [0011] 일 실시 예에 있어서, 방법은 호스트에 연결된 제 2 그래픽 프로세싱 유닛에 의해, 키를 포함하는 제 2 키 값 요청을 제 2 그래픽 프로세싱 유닛에 연결된 제 2 메모리의 입력-출력 영역 내 키 값 요청 큐에 쓰는 단계; 호스트 및 제 2 메모리에 연결된 제 2 키 값 스토리지 장치에 의해, 제 2 키 값 요청을 키 값 요청 큐로부터 읽는 단계, 및 제 2 키 값 스토리지 장치에 의해, 제 2 키 값 요청에 응답하여, 제 2 키 값 요청의 키에 대응하는 제 2 값을 제 2 메모리의 입력-출력 영역에 쓰는 단계를 포함한다.
- [0012] 일 실시 예에 있어서, 방법은: 제 1 키 값 스토리지 장치에 의해, 제 1 키 값 스토리지 장치에서, 키 검색을 수행하여 제 1 값을 회수하는 단계; 및 제 1 키 값 스토리지 장치에 의한 키 값 검색을 수행하는 단계와 동시에, 제 2 키 값 스토리지 장치에 의해, 제 2 키 값 스토리지 장치에서, 키 검색을 수행하여 제 2 값을 회수하는 단계를 포함한다.
- [0013] 일 실시 예에 있어서, 제 1 키 값 스토리지 장치에 의해, 제 1 키 값 요청을 읽는 단계는 P2P(peer-to-peer) DMA(direct memory access)를 통해 제 1 키 값 요청을 읽는 단계를 포함한다.
- [0014] 일 실시 예에 있어서, 제 1 키 값 스토리지 장치에 의해, 제 1 값을 쓰는 단계는 P2P(peer-to-peer) DMA(direct memory access)를 통해 제 1 값을 쓰는 단계를 포함한다.
- [0015] 일 실시 예에 있어서, 제 1 키 값 스토리지 장치는 PCI(peripheral component interconnect) 연결에 의해 제 1 그래픽 프로세싱 유닛에 연결된다.
- [0016] 일 실시 예에 있어서, 방법은 제 1 그래픽 프로세싱 유닛에 의해, 제 1 키 값 요청을 쓰는 단계 이후 그리고 제 1 키 값 스토리지 장치에 의해, 쓰기 단계에 의해, 제 1 값을 쓰는 단계 이전에, 제 1 그래픽 프로세싱 유닛에 의해, 제 2 키 값 요청을 키 값 요청 큐에 쓰는 단계를 포함한다.
- [0017] 일 실시 예에 있어서, 방법은: 제 1 그래픽 프로세싱 유닛에 의해, 키를 포함하는 제 2 키 값 요청을 제 1 메모리의 제 2 입력-출력 영역 내 키 값 요청 큐에 쓰는 단계; 제 1 메모리에 연결된 제 2 키 값 스토리지 장치에 의해, 제 2 키 값 요청을 제 1 메모리의 제 2 입력-출력 영역의 키 값 요청 큐로부터 읽는 단계, 및 제 2 키 값 스토리지 장치에 의해, 제 2 키 값 요청에 응답하여, 제 2 키 값 요청의 키에 대응하는 제 2 값을 제 1 메모리의 제 2 입력-출력 영역에 쓰는 단계를 포함한다.
- [0018] 일 실시 예에 있어서, 방법은: 제 1 키 값 스토리지 장치에 의해, 제 1 키 값 스토리지 장치에서, 키 검색을 수행하여 제 1 값을 회수하는 단계, 및 제 1 키 값 스토리지 장치에 의한 키 값 검색을 수행하는 단계와 동시에, 제 2 키 값 스토리지 장치에 의해, 제 2 키 값 스토리지 장치에서, 키 검색을 수행하여 제 2 값을 회수하는 단계를 포함한다.
- [0019] 본 발명의 실시 예에 따라 머신 러닝을 위한 시스템이 제공되고, 시스템은: 그래픽 프로세싱 유닛; 그래픽 프로세싱 유닛에 연결된 메모리; 및 키 값 스토리지 장치를 포함하고; 키 값 스토리지 장치는 PCI(peripheral component interconnect) 연결에 의해 그래픽 프로세싱 유닛에 연결되고; 그래픽 프로세싱 유닛은 메모리의 입력-출력 영역 내 메모리가 매핑(mapping)된 입력 및 출력 동작들을 수행하고, 그리고 하나 이상의 키 값 요청들을 입력-출력 영역 내 키 값 요청 큐에 쓰도록 구성되고; 키 값 스토리지 장치는: 입력-출력 영역에서 메모리가 매핑된 입력 및 출력 동작들을 수행하고; 하나 이상의 키 값 요청들을 키 값 요청 큐로부터 읽고; 그리고 하나 이상의 키 값 요청들의 키 값 요청에 응답하여, 키 값 요청의 키에 대응하는 값을 메모리의 입력-출력 영역에 쓰도록 구성된다.
- [0020] 일 실시 예에 있어서, 키 값 요청은 값에 할당된 영역인 리턴-값 영역을 포함한다.

- [0021] 일 실시 예에 있어서, 값을 메모리의 입력-출력 영역에 쓰는 단계는 값을 리턴-값 영역에 쓰는 단계를 포함한다.
- [0022] 일 실시 예에 있어서, 값을 메모리의 입력-출력 영역에 쓰는 단계는 값을 메모리의 입력-출력 영역 내 리턴-값 큐에 쓰는 단계를 포함한다.
- [0023] 본 발명의 실시 예에 따라 머신 러닝을 위한 시스템이 제공되고, 시스템은: 그래픽 프로세싱 유닛; 키 값 스토리지 장치; 및 그래픽 프로세싱 유닛과 키 값 스토리지 장치 간의 통신을 위한 공유된 메모리 수단을 포함하고; 그래픽 프로세싱 유닛은 통신을 위한 공유된 메모리 수단을 통해 하나 이상의 키 값 요청들을 키 값 스토리지 장치로 전송하도록 구성되고, 키 값 스토리지 장치는: 하나 이상의 키 값 요청들을 수신하고; 그리고 하나 이상의 키 값 요청들의 키 값 요청에 응답하여, 키 값 요청의 키에 대응하는 값을 통신을 위한 공유된 메모리 수단을 통해 그래픽 프로세싱 유닛에 전송하도록 구성된다.
- [0024] 일 실시 예에 있어서, 통신을 위한 공유된 메모리 수단은 그래픽 프로세싱 유닛에 연결된 메모리를 포함하고, PCI(peripheral component interconnect) 연결을 통한 P2P(peer-to-peer) DMA(direct memory access)를 통해 키 값 스토리지 장치에 의해 접근되도록 구성된다.

### 발명의 효과

- [0025] 본 발명의 실시 예에 따르면, 호스트 어플리케이션에 의해 수행되는 작업(task)은 단지 GPU와 SSD 간의 통신을 위한 경로를 설정하는 것을 포함하고, GPU 계산의 직렬화가 방지될 수 있다. 따라서 본 발명의 실시 예들은 규모 확장(scale out) 다수의 GPU들을 가능하게 할 수 있어 머신 러닝 트레이닝을 가속화한다.

### 도면의 간단한 설명

- [0026] 본 발명의 이들 및 다른 특징들 및 이점들은 발명을 실시하기 위한 구체적인 내용, 청구 범위, 및 첨부 도면을 참조하여 인식되고 이해될 것이다:
- 도 1은 머신 러닝을 위한 시스템의 기능적 블록도이다;
- 도 2는 본 발명의 실시 예에 따른, 온보드 SSD를 갖춘 그래픽 카드의 블록도이다;
- 도 3은 본 발명의 실시 예에 따른, 데이터 흐름도이다;
- 도 4는 본 발명의 실시 예에 따른, 타이밍도이다; 그리고
- 도 5는 본 발명의 실시 예에 따른, 타이밍도이다.

### 발명을 실시하기 위한 구체적인 내용

- [0027] 첨부된 도면들과 관련되어 이하에서 제공되는 상세한 설명은 본 발명에 따라 제공되는 데이터에 대한 키 값 접근을 포함하는 머신 러닝을 수행하기 위한 시스템 및 방법의 예시적인 실시 예들의 설명으로서 의도되고 본 발명이 구성될 수 있거나 활용될 수 있는 유일한 형태들을 나타내는 것으로 의도되지 않는다. 설명은 도시된 실시 예들과 관련된 본 발명의 특징들을 제공한다. 그러나, 동일하거나 동등한 기능들 및 구조들은 본 발명의 범위 내에 포함되는 것으로도 의도되는 다른 실시 예들에 의해 달성될 수 있다. 본문의 다른 곳에서도 표시된 바와 같이, 유사한 구성 요소 번호들은 유사한 구성 요소들 또는 특징들을 나타내는 것으로 의도된다.
- [0028] 종래 기술의 머신 러닝(machine learning) 플랫폼(platform)들은 전체 트레이닝(training) 데이터의 부분 집합을 랜덤하게(randomly) 샘플링하는 확률적(stochastic) 머신 러닝 트레이닝에 이용되는 경우에 단점을 갖는다. 이러한 머신 러닝 플랫폼들은 확률적 머신 러닝 트레이닝 동안 키 값 접근으로 인한 낮은 GPU(graphics processing unit; 그래픽 프로세싱 유닛) 활용으로 문제를 겪을 수 있고, 이는 CPU(central processing unit)를 포함하여 PCIe(peripheral component interconnect express) 버스를 가로질러(traversing) 키 값 인터페이스 및 데이터 전송을 제공하는 것을 요구하기 때문이다. 전술한대로, 일부 종래 기술의 시스템들에서, 호스트 CPU는 파일 인덱스 검색(lookup) 및 파일 시스템 접근을 수행하여 데이터의 위치를 식별하고, 이것은 직렬화된 키 값 접근으로 이어진다. 반면에, 일부 실시 예들에서, CPU가 온보드(onboard) SSD(solid state drive)에 저장된 데이터에 대한 키 값 접근에서는 관여하지 않으므로 성능은 향상된다. GPU는 온보드 키 값 스토리지 장치(예를 들어, 온보드 키 값 SSD), 예를 들어, 비동기(asynchronous) 키 값 접근을 가능하게 하여 접근 대기 시간(latency)의 효과를 줄이는, GPU 및 온보드 키 값 SSD를 포함하는 온 그래픽 카드(on a graphics card)로 키



값 명령들을 직접 전송한다. 본문에서 사용되는 것으로서, “키 값 스토리지 장치”는 각 이러한 요청에 응답하여 요청에 포함된 키에 대응하는 값을 되돌려줌으로써 키 값 요청들(각각은 키를 포함)에 응답하도록 구성되는(SSD와 같은) 영구 스토리지 장치이다.

[0029] 도 1은 GPU 머신 러닝 트레이닝 동안, 소프트웨어 키 값 저장을 이용하여, 블록 인터페이스를 갖는 SSD 상에 저장된 데이터에 대한 키 값 접근의 전체적인 흐름을 도시한다. 먼저, 호스트 어플리케이션(105)은 “GET “ 요청(혹은 얻기 요청)을 소프트웨어 키 값 저장(110; S/W KV Store, 소프트웨어 키 값 저장소로도 지칭될 수 있음)으로 전송함으로써 키 값 접근을 시작한다. 어떤 키에 대한 데이터의 위치를 식별하기 위해, 소프트웨어 키 값 저장(110)은 지정된(specified) 키에 대응하는 데이터의 파일 오프셋(offset)을 저장하는 인덱스 표에 접근한다. 그러면 소프트웨어 키 값 저장(110)은 파일 오프셋을 갖는 파일 시스템(115)에 접근하고, 그리고 파일 시스템(115)은 블록 인터페이스(120)를 갖는 SSD에 접근하고 지정된 키에 대응하는 데이터를 페치(fetch)한다. 일단 값이 호스트 어플리케이션에게 사용할 수 있게 되면, 호스트 어플리케이션은 GPU 계산을 위한 GPU 메모리(125)로 값을 전송하고 GPU 커널(kernel)을 시작한다(125). 블록 인터페이스를 갖는 종래 기술의 SSD를 갖는 소프트웨어 키 값 저장을 위해, 모든 이들 동작들은 순차적으로(sequentially) 수행된다.

[0030] 도 1에서 도시된 바와 같이, 블록 인터페이스를 갖는 SSD에 대한 키 값 접근은 지정된 키에 대응하는 데이터의 위치를 식별하기 위해 호스트 상에서 실행되는 몇몇의 계산 단계들을 포함한다. GPU는 소프트웨어 키 값 저장의 계산들이 끝난 이후에만 해당 계산을 수행할 수 있다. GPU들이 다른 GPU들로부터의 키 값 접근들의 완료를 기다려야 하고 이는 GPU 계산들의 직렬화로 이어지기 때문에 더 많은 GPU들이 시스템에서 사용되는 경우 키 값 접근의 대기 시간은 증가한다. 결과적으로, 블록 인터페이스를 갖는 종래 기술의 SSD들에 대한 키 값 접근은 시스템에서 생산적으로 사용될 수 있는 GPU들의 개수를 제한한다.

[0031] 일부 실시 예들에서, 키 값 인터페이스(또는 “키 값 SSD”)를 갖는 온보드 SSD를 갖는 그래픽 카드가 종래 기술의 시스템들의 단점의 일부를 극복하는데 사용된다. 도 2는 이러한 장치를 갖는 예시적인 시스템을 도시한다. 스토리지 및 GPU 간의 데이터 이동이 글로벌 PCIe 버스를 가로지르는 데이터 전송을 요구하고, 이는 GPU로부터의 데이터 접근들의 대기 시간을 증가시키는, 종래 기술의 시스템과 달리, 온보드 키 값 SSD(205)를 갖는 그래픽 카드는 온보드 키 값 SSD(205) 및 GPU(210) 간의 P2P(peer to peer) DMA(direct memory access)를 이용함으로써 오버헤드(overhead)를 줄일 수 있고 완전한 P2P DMA 제어를 GPU(210)에게 준다. 일부 실시 예들에서, 온보드 키 값 SSD(205)는 비표준적인(non-standard) 명령들로서 키 값 명령들을 제공한다. 예를 들어, 키 값 요청 큐(queue, 이하에서 더 상세하게 설명됨)는 NVMe(nonvolatile memory express) 명령 큐로서 구현될 수 있고, NVMe 명령들은, SSD(205)의 펌웨어(firmware)에서 그리고 GPU 상의 드라이버 소프트웨어에서, 정의된 공급업체별(vendor-specific) 명령들이며, 키 값 명령들, 즉, SSD(205)로부터, 명령의 일부로서 제공되는 키에 대응하는 값을 요청하는데 사용되는 명령들에 대응한다.

[0032] 일부 실시 예들에서, 이러한 시스템은 온보드 키 값 SSD에서 비동기 키 값 접근을 제공하는데 사용될 수 있고, 일부 실시 예들은 트레이닝 데이터의 랜덤 샘플링을 위해 그래픽 카드 내에서 키 값 SSD를 활용한다. 도 3은 일부 실시 예들에서, 머신 러닝 트레이닝 동안 키 값 접근의 흐름을 도시한다. 이러한 실시 예와 일부 종래 기술의 시스템들 간의 중요한 차이는 이러한 실시 예에서 GPU는 키 값 명령들을 키 값 SSD(205)로 직접 전송한다는 것이다. 먼저, 호스트 어플리케이션의 실행의 초기 단계에서, 호스트 어플리케이션은 특정한 GPU 장치의 메모리를 PCI(peripheral component interconnect) BAR(base address registers) 메모리 영역으로 매핑(mapping)하여 키 값 SSD(205) 및 GPU 간의 직접적인 통신을 설정(establish)한다. 이 과정에 의해, 키 값 SSD(205) 및 GPU 간의 통신(예를 들어, 메모리가 매핑(mapping)된 입력 출력에 의해)에 할당된 GPU 메모리의 영역은 GPU 메모리의 “입력-출력 영역”으로서 본문에서 지칭될 수 있다. GPU 및 키 값 SSD(205) 모두에 의해 직접적으로 접근 가능한, GPU 메모리의 입력-출력 영역은 공유된 메모리로서 기능적으로 동작할 수 있다. GPU 어플리케이션(305)은 키 값 SSD(205) 상에서 메모리가 매핑된 IO(입력-출력)를 수행하고 노출된 GPU 메모리의 버스 어드레스를 공급함으로써 키 값 SSD(205)로 GET 요청을 발행한다. 키 값 SSD(205) 내에서 펌웨어는 키 검색을 수행하여 키에 대응하는 값을 회수한 이후에, 호스트 어플리케이션(105)의 중재(intermediation) 없이 그 값을 매핑된 GPU 장치 메모리에(즉, GPU 메모리의 입력-출력 영역에) 쓴다.

[0033] 일부 실시 예들에서, GPU가 두 번째, 다음의 요청을 만들기 전에 첫 번째 요청에 대한 응답을 기다릴 필요가 없다는 의미에서, 키 값 요청 큐(key value request queue; KVRQ, 310)가 사용되고, 키 값 접근은 비차단적(non-blocking)이다. 대신에, GPU는 키 값 요청들을 키 값 요청 큐(310)에 배치하고, 그리고 요청들은 키 값 SSD(205)에 의해 차례로 처리된다. 이와 같이, GPU 어플리케이션이 요청을 키 값 요청 큐(310)에 넣으면 요청 동작이 완료된다. 키 값 요청 큐(310)의 엔트리들의 개수가 키 값 요청들의 개수이도록 키 값 요청 큐(310)는



완료되지 않은 요청들을 가지고 있다(hold). SSD(205) 내의 펌웨어는 값이 GPU 메모리로 전송되는 경우 지정된 키에 대응하는 키 값 요청 큐 엔트리를 방출한다(release).

[0034] 각 GPU에 대한 분리된 키 값 접근은 다수의 GPU들로부터의 키 값 접근을 겹치게 하는(중첩하는) 것을(또는 GPU들로부터의 오버래핑(overlapping) 키 값 접근을) 가능하게 한다. 예를 들어, 두 개의 GPU들을 갖는 시스템에서, 각 GPU는 각각의 키 값 SSD에 연결되고, 두 개의 GPU들은 동시에 요청들을 발행하고, 그리고 그들 각각의 키 값 SSD들은 동시에 응답할 수 있다. 도 4는 각각의 키 값 SSD에 각각 연결된 두 개의 GPU들을 포함하는 이러한 예시의 동작을 도시한다. 도 4는 또한 GPU 계산이 직렬화된 종래 기술의 접근 방법과 비교하여, 키 값 접근들을 겹치게 하는 것이 두 개의 GPU들로 수행되는, 이러한 실시 예에서, 시스템으로 절약된 시간을 도시한다. 일부 실시 예들에서, 3개 이상의 (예를 들면, 임의의 개수) GPU들은 각각의 키 값 SSD에 각각 연결될 수 있고 키 값 동작들을 (예를 들면, 동시에) 겹치게 하는 것을 수행한다.

[0035] 일부 실시 예들에서, 키 값 접근을 위한 요청 및 응답의 분리는 비동기 키 값 접근을 가능하게 하고, 예를 들어, GPU로부터 다수의 요청들의 배치(batching)이 가능하다. 도 5는 두 개 키 값 명령들을 배치하는 경우 비동기 키 값 접근들의 예시를 도시한다. GPU 계산 및 SSD 장치 접근이 직렬화되는 동기 키 값 접근과 비교하여, 일부 실시 예들의 비동기 키 값 접근은 GPU 계산을 갖는 다수의 키 값 명령들의 겹침(오버래핑)을 가능하게 한다. 이 예시에서, GPU는 이전 요청의 완료를 위해, 그 때마다, 기다리는 것 대신에 GET 요청들을 연속적으로 발행한다. 일부 실시 예들에서, 3개 이상의 (예를 들면, 임의의 개수) GPU들은 각각의 키 값 SSD에 각각 연결될 수 있고 키 값 동작들을 (예를 들면, 동시에) 겹치게 하는 것을 수행한다.

[0036] 일부 실시 예들에서, 키 값 SSD가 키 값 요청에 응답하여 값을 회수하는 경우, 그것은 키 값 요청 큐에, 즉 이 목적으로 키 값 요청 내에 할당된 메모리의 영역(또는 “리턴-값(return-value) 영역”)에 다시 회수된 값을 쓴다. 다른 실시 예들에서, 키 값 SSD는 GPU 메모리의 입력-출력 영역에 할당된, 분리된 큐 (또는 “리턴 값 큐”)에 회수된 값을 대신 쓴다. 일부 실시 예들에서, 각 GPU가 그것이 키-값 요청들을 전송하는 하나의 전용의 키 값 SSD를 갖는 것 대신에, 하나의 GPU는 여러 키 값 SSD들을 가질 수 있다. 이러한 실시 예에서, 여러 키 값 요청 큐들은 각각의 키 값 SSD에 대해 각각 GPU 메모리 내에 할당될 수 있다. 다른 실시 예들에서, 여러 GPU들은 하나의 키 값 SSD에 연결될 수 있고, 하나의 키 값 SSD는, 예를 들어, 라운드-로빈(round-robin) 방식으로, GPU들의 각각의 키 값 요청 큐들에서, 키 값 요청들을 제공할 수 있다.

[0037] 일부 실시 예들에서, 호스트 어플리케이션에 의해 수행되는 작업(task)은 단지 GPU와 SSD 간의 통신을 위한 경로를 설정하는 것을 포함하고, GPU 계산의 직렬화를 방지함으로써 이들 실시 예들의 확장성(scalability)을 향상시키고 GPU 계산의 직렬화가 방지되지 않으면 CPU 상의 호스트 어플리케이션에 의해 수행되는 키 값 접근 동작들이 발생할 수 있다. 이와 같이, 이들 실시 예들은 규모 확장(scale out) 다수의 GPU들이 머신 러닝 트레이닝을 가속화하게 할 수 있다. 복잡한 키 값 소프트웨어를 간단한 장치 인터페이스로 교체함으로써, 일부 실시 예들은 또한, 그렇지 않으면 호스트 상에 부과될 수 있는, CPU 코어들의 개수에 대한 요구를 포함하는 자원 요구들을 줄인다. 이러한 요구를 방지하는 것은 더 나은 에너지 효율을 야기할 수 있다.

[0038] 일부 실시 예들은 하나 이상의 프로세싱 회로들을 이용하여 구성될 수 있다. 본문에서 사용된 “프로세싱 회로” 용어는 데이터 또는 디지털 신호들을 처리하기 위해 이용되는, 하드웨어, 펌웨어, 및 소프트웨어의 임의의 조합을 의미한다. 프로세싱 회로 하드웨어는 예를 들어, ASICs(application specific integrated circuits), 일반적인 목적의 또는 특수한 목적의 CPUs(central processing units), DSPs(digital signal processors), GPUs(graphics processing units), 및 FPGAs(field programmable gate arrays)와 같은 프로그래머블(programmable) 로직 장치들을 포함할 수 있다. 프로세싱 회로에서, 본문에서 사용된 바와 같이, 각 기능은 기능을 수행하기 위해 구성된 하드웨어(즉, 하드웨어에 내장(hard-wired)), 또는 비일시적(non-transitory) 저장 매체에 저장된 명령들을 실행하도록 구성된, CPU와 같은, 좀 더 일반적인 목적의 하드웨어 중 어느 하나에 의해 수행된다. 프로세싱 회로는 하나의 PCB(printed circuit board) 상에서 제조될 수 있거나 또는 상호 연결된 여러 PCB들에 걸쳐 분산될 수 있다. 프로세싱 회로는 다른 프로세싱 회로들을 포함할 수 있다; 예를 들어, 프로세싱 회로는 PCB 상에 상호 연결된, 두 개의 프로세싱 회로들, FPGA, 및 CPU를 포함할 수 있다.

[0039] 다양한 구성 요소들, 성분들, 영역들, 계층들, 및/또는 섹션들을 설명하기 위해 “제 1”, “제 2”, “제 3”, 등이 본문에서 사용될 수 있지만, 이들 구성 요소들, 성분들, 계층들, 및/또는 섹션들은 이들 용어들에 의해 제한되지 않는 것으로 이해될 것이다. 이들 용어들은 단지 하나의 구성 요소, 성분, 영역, 계층, 또는 섹션으로부터 구별하기 위해 사용된다. 따라서, 본문에서 논의된 제 1 구성 요소, 성분, 영역, 계층, 또는 섹션은 본 발명의 사상 및 범위를 벗어나지 않고 제 2 구성 요소, 성분, 영

역, 계층, 또는 섹션으로 지칭될 수 있다.

[0040] 본문에서 사용된 용어들은 특정한 실시 예들을 설명하기 위한 것이고 본 발명을 제한하려고 의도된 것이 아니다. 본문에서 사용된 바와 같이, “실질적으로”, “약” 용어들 및 유사한 용어들은 근사치의 용어들로서 사용되고 정도의 용어들로서 사용되지 않고, 당업자에 의해 식별되는 측정된 또는 계산된 값들의 고유한 변동들을 고려하기 위한 것이다. 본문에서 사용된 바와 같이, “주(major) 성분” 용어는 조성물, 중합체, 또는 조성물 또는 생성물의 임의의 다른 하나의 성분의 양보다 큰 양의 생성물에 존재하는 성분을 지칭한다. 반면에, “주(primary) 성분” 용어는 조성물, 중합체, 또는 생성물의 중량의 적어도 50% 이상을 구성하는 성분을 나타낸다. 본문에서 사용된 바와 같이, “주 부분” 용어는 복수의 항목들에 적용될 때, 항목들의 적어도 절반을 의미한다.

[0041] 본문에서 사용된 바와 같이, 문맥상 명백하게 다르게 나타내지 않는 한, 단수 표현들은 복수의 형태들도 포함하는 것으로 의도된다.” 포함” 및/또는 “포함하는” 용어들은, 본 명세서에서 사용되는 경우, 기술된 특징들, 숫자들, 단계들, 동작들, 구성 요소들, 및/또는 성분들의 존재를 명시하지만 하나 이상의 다른 특징들, 숫자들, 단계들, 동작들, 구성 요소들, 성분들, 및/또는 이들의 그룹들의 존재 또는 부가를 배제하지 않는 것으로 더 이해될 것이다. 본문에서 사용된 바와 같이, 용어 “및/또는”은 관련된 열거된 항목들의 하나 이상의 임의의 그리고 모든 조합들을 포함한다.” 적어도 하나의”와 같은 표현들은, 요소들의 리스트에 앞서는 경우, 요소들의 전체 리스트를 수정하고 리스트의 개별 요소들을 수정하지 않는다. 또한, “할 수 있는”의 사용은 본 발명의 실시 예들을 설명하는 경우 본 발명의 하나 이상의 실시 예들”을 지칭한다. 또한, “예시적인” 용어는 예시 또는 도시를 지칭하는 것으로 의도된다. 본문에서 사용된 바와 같이, “이용”, “이용하는”, 및 “이용된” 용어들은 “활용”, “활용하는”, 및 “활용된” 용어들과 같은 것을 의미하는 것으로 간주될 수 있다.

[0042] 구성 요소 또는 계층이 다른 구성 요소 또는 계층 “상의”, “연결된”, 또는 “인접한”으로서 지칭되는 경우, 그것은 직접적으로 다른 구성 요소 또는 계층 상의, 연결된, 또는 인접한 것일 수 있거나, 하나 이상의 중간 구성 요소들 또는 계층들이 존재할 수 있음이 이해될 것이다. 반면에, 구성 요소 또는 계층이 다른 구성 요소 또는 계층 “에 직접적으로”, “에 직접적으로 연결된”, “의 바로 인접하여”로서 지칭되는 경우, 중간 요소들 또는 계층들은 존재하지 않는다.

[0043] 데이터로의 키 값 접근을 포함하는 머신 러닝을 수행하기 위한 시스템 및 방법의 예시적인 실시 예들이 구체적으로 설명되었고 본문에서 도시되었으나, 많은 수정들 및 변형들은 당업자에게 분명할 것이다. 따라서, 본 발명의 원리들에 따라 구성된 데이터로의 키 값 접근을 포함하는 머신 러닝을 수행하기 위한 시스템 및 방법은 구체적으로 본문에서 설명된 것과 달리 구현될 수 있다. 본 발명은 다음의 청구 범위, 및 그것의 균등물에서 또한 정의된다.

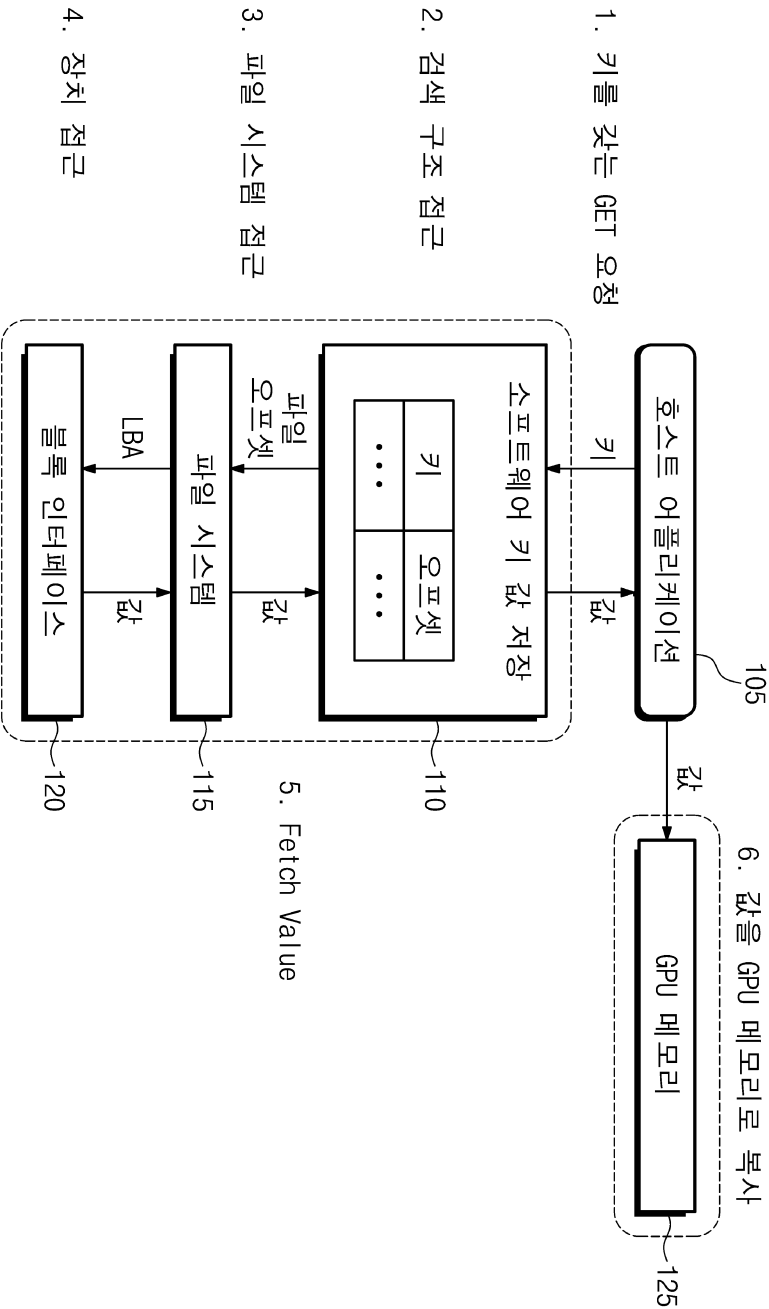
### 부호의 설명

[0044]

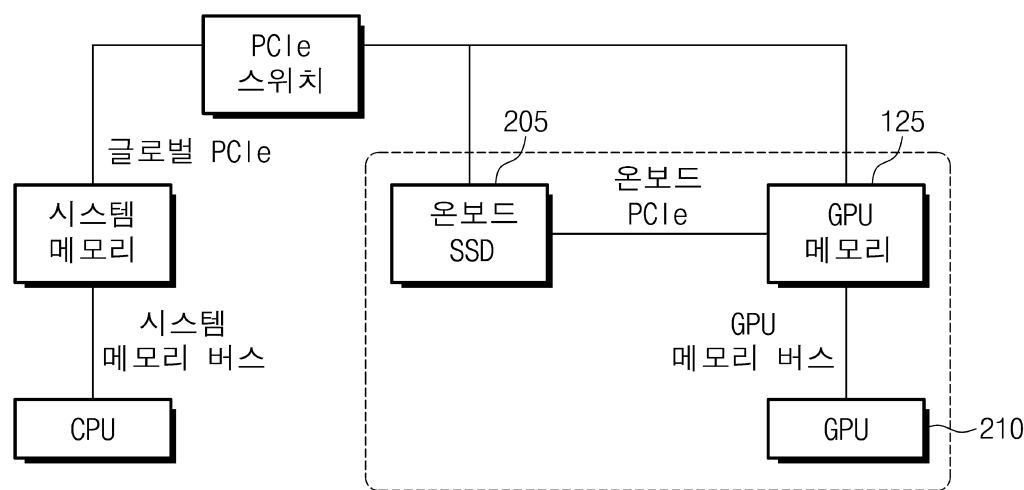
105: 호스트 어플리케이션;	110: 소프트웨어 키 값 저장;
115: 파일 시스템;	120: 블록 인터페이스;
125: GPU 메모리;	205: 온보드 키 값 SSD;
210: GPU;	305: GPU 어플리케이션;
310: 키 값 요청 큐;	

도면

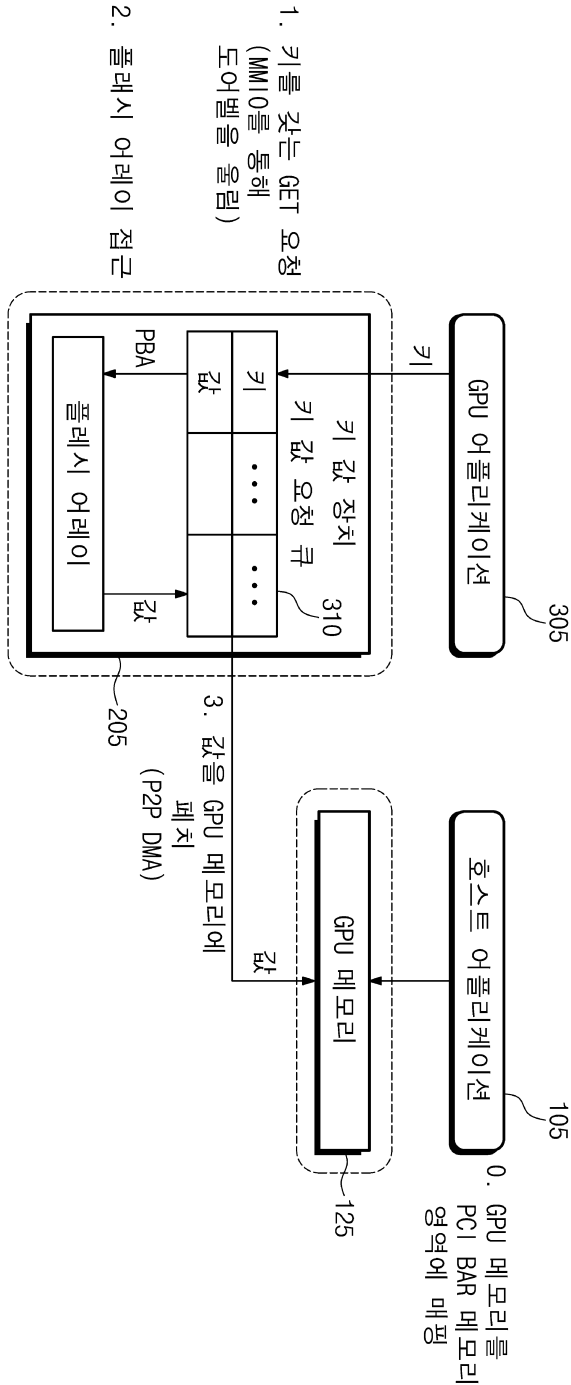
도면1



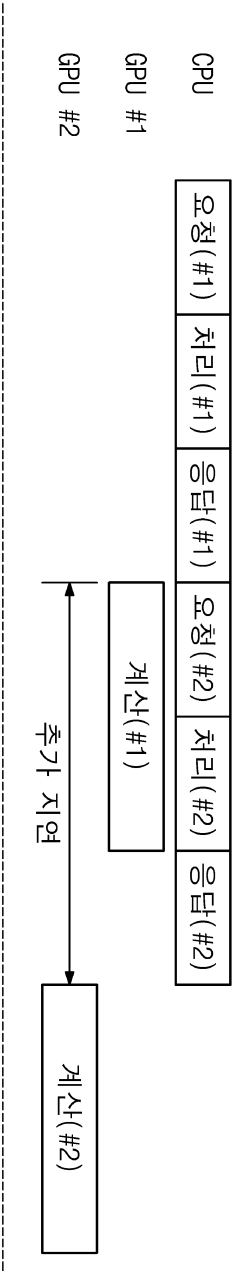
도면2



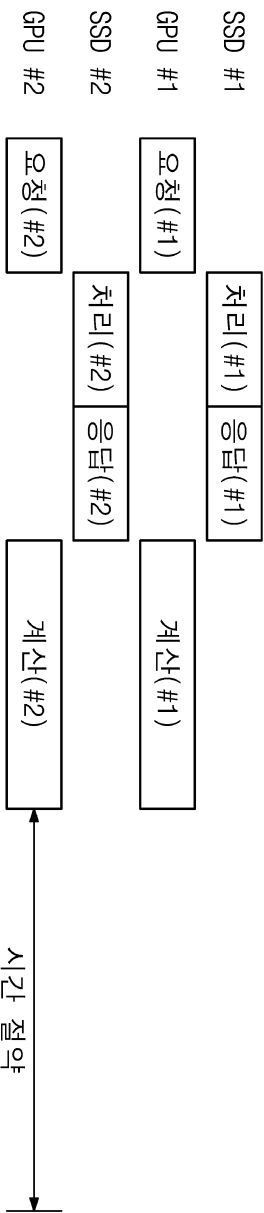
도면3



소프트웨어 키 값 저장



키 값 SSD



도면4



도면5

