(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년04월03일
(11) 등록번호 10-2516184
(24) 등록일자 2023년03월27일

(51) 국제특허분류(Int. Cl.)
G06F 3/06 (2006.01) *G06N 3/02* (2023.01)
(52) CPC특허분류
G06F 3/061 (2013.01)
G06F 3/0683 (2013.01)
(21) 출원번호 10-2022-0119113
(22) 출원일자 2022년09월21일
심사청구일자 2022년09월21일
(56) 선행기술조사문헌
KR1020170027036 A*
KR1020180072345 A*
KR102425909 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
(주)글루시스
경기도 안양시 동안구 시민대로327번길 11-31 , 5층(관양동, 파낙스알앤디센터)
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
에프에이리눅스 주식회사
경기도 의왕시 이미로 40, 비동 1001호 1002호 (포일동, 인덕원아이티밸리)
(72) 발명자
노재춘
서울특별시 광진구 능동로 209, 세종대학교 대양AI센터 442호(군자동)
김정명
경기도 광명시 양지로 7, 104동 2104호(일직동, 유-플래닛 광명역 데시앙)
(뒷면에 계속)
(74) 대리인
민영준

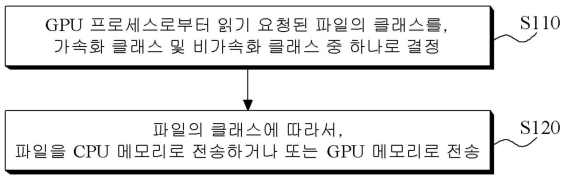
전체 청구항 수 : 총 8 항 심사관 : 정성훈

(54) 발명의 명칭 **GPU 데이터 입출력 가속화 방법**

(57) 요 약

GPU의 데이터 입출력을 가속화시키는 방법이 개시된다. 개시된 GPU 데이터 입출력 가속화 방법은 GPU 프로세스로부터 읽기 요청된 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계; 및 상기 파일의 클래스에 따라서, 상기 파일을 CPU 메모리로 전송하거나 또는 GPU 메모리로 전송하는 단계를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06N 3/02 (2023.01)
G06F 2212/1016 (2013.01)
(72) 발명자
피아오웬유
서울특별시 광진구 광나루로22나길 12-1
박성순
경기도 군포시 수리산로 244, 995동 901호(산본동, 한양백두아파트)

박문식
경기도 부천시 역곡로85번길 3(역곡동)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711153020
과제번호	2020-0-00104-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	차세대엣지컴퓨팅시스템기술개발(R&D)
연구과제명	저지연 입출력 집약적 엣지 데이터 처리를 위한 스토리지 모듈 기술 개발
기 여 율	1/1
과제수행기관명	(주)글루시스
연구기간	2022.01.01 ~ 2022.12.31

명 세 서

청구범위

청구항 1

컴퓨팅 장치에 의해 수행되는, GPU 데이터 입출력 가속화 방법에 있어서,

GPU 프로세스로부터 읽기 요청된 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계; 및

상기 파일의 클래스에 따라서, 상기 파일을 CPU 메모리로 전송하거나 또는 GPU 메모리로 전송하는 단계를 포함하며,

상기 파일을 CPU 메모리로 전송하거나 또는 GPU 메모리로 전송하는 단계는

상기 파일의 클래스가 상기 가속화 클래스인 경우, 제1저장 장치가 마운트된 제1디렉토리에 상기 파일을 저장하여, 상기 GPU 메모리로 전송하며

상기 파일의 클래스가 상기 비가속화 클래스인 경우, 상기 제1저장 장치보다 입출력 속도가 느린 제2저장 장치가 마운트된 제2디렉토리에 상기 파일을 저장하여, 상기 CPU 메모리로 전송하며

상기 제2디렉토리는, CPU 프로세스로부터 읽기 요청된 파일이 저장된 디렉토리이며,

상기 GPU 데이터 입출력 가속화 방법은

상기 제2디렉토리에 저장된 파일에 할당된 메모리가 GPU 메모리인지 여부를 판단하여, 상기 제2디렉토리에 저장된 파일에 대한 읽기 요청을 상기 CPU 프로세스에 의한 요청 또는 상기 GPU 프로세스에 의한 요청으로 분류하는 단계; 및

상기 분류 결과에 따라 상기 제2디렉토리에 저장된 파일을, 상기 CPU 메모리를 경유시켜, 상기 GPU 메모리로 전송하는 단계

를 더 포함하는 GPU 데이터 입출력 가속화 방법.

청구항 2

제 1항에 있어서,

상기 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계는

상기 파일에 대한 제1특징 정보 및 상기 GPU 프로세스가 실행중인 시스템에 대한 제2특징 정보를 수집하는 단계; 및

상기 제1 및 제2특징 정보를 미리 학습된 분류 모델에 입력하여, 상기 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계

를 포함하는 GPU 데이터 입출력 가속화 방법.

청구항 3

제 2항에 있어서,

상기 제1특징 정보는

상기 파일의 크기 정보를 포함하며,

상기 제2특징 정보는

CPU 및 상기 GPU의 사용량, 상기 CPU 메모리 및 GPU 메모리의 크기, 시스템의 프레임워크 정보, 입출력 접근 패턴, 입출력 세분성 정보 중 적어도 하나를 포함하는

GPU 데이터 입출력 가속화 방법.

청구항 4

삭제

청구항 5

제 1항에 있어서,

상기 파일을 CPU 메모리로 전송하거나 또는 GPU 메모리로 전송하는 단계는

상기 제1디렉토리에 저장된 파일을, 상기 GPU 메모리와 상기 제1저장 장치 사이의 직접 경로를 통해, 상기 GPU 메모리로 전송하며,

상기 제2디렉토리에 저장된 파일을, 상기 CPU 메모리로 전송하는,

GPU 데이터 입출력 가속화 방법.

청구항 6

삭제

청구항 7

삭제

청구항 8

제 1항에 있어서,

상기 제1저장 장치는, NVMe SSD이며,

상기 제2저장 장치는, HDD인

GPU 데이터 입출력 가속화 방법.

청구항 9

컴퓨팅 장치에 의해 수행되는, GPU 데이터 입출력 가속화 방법에 있어서,

GPU 프로세스로부터 읽기 요청된 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계; 및

상기 파일의 클래스에 따라서, 입출력 속도가 서로 다른 저장 장치에 마운트된 제1 및 제2디렉토리 중 하나에 상기 파일을 저장하는 단계를 포함하며,

상기 제1 및 제2디렉토리 중 하나에 상기 파일을 저장하는 단계는

상기 파일의 클래스가 상기 가속화 클래스인 경우, 제1저장 장치가 마운트된 제1디렉토리에 상기 파일을 저장하고,

상기 파일의 클래스가 상기 비가속화 클래스인 경우, 상기 제1저장 장치보다 입출력 속도가 느린 제2저장 장치가 마운트된 제2디렉토리에 상기 파일을 저장하며

상기 제2디렉토리는, CPU 프로세스로부터 읽기 요청된 파일이 저장된 디렉토리이며,

상기 제1디렉토리에 저장된 파일은, GPU 메모리와 상기 제1저장 장치 사이의 직접 경로를 통해, 상기 GPU 메모리로 전송되며,

상기 제2디렉토리에 저장된 파일은, CPU 메모리 또는 상기 GPU 메모리로 전송되며,

상기 GPU 데이터 입출력 가속화 방법은

상기 제2디렉토리에 저장된 파일에 할당된 메모리가 GPU 메모리인지 여부를 판단하여, 상기 제2디렉토리에 저장된 파일에 대한 읽기 요청을 상기 CPU 프로세스에 의한 요청 또는 상기 GPU 프로세스에 의한 요청으로 분류하는 단계; 및

상기 분류 결과에 따라 상기 제2디렉토리에 저장된 파일을 상기 CPU 메모리를 경유시켜, 상기 GPU 메모리로 전송하는 단계

를 더 포함하는 GPU 데이터 입출력 가속화 방법.

청구항 10

제 9항에 있어서,

상기 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계는

상기 파일에 대한 제1특징 정보 및 상기 GPU 프로세스가 실행중인 시스템에 대한 제2특징 정보를 수집하는 단계; 및

상기 제1 및 제2특징 정보를 미리 학습된 분류 모델에 입력하여, 상기 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계

를 포함하는 GPU 데이터 입출력 가속화 방법.

청구항 11

삭제

청구항 12

삭제

청구항 13

제 9항에 있어서,

상기 제1저장 장치는, NVMe SSD이며,

상기 제2저장 장치는, HDD인

GPU 데이터 입출력 가속화 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 GPU의 데이터 입출력을 가속화시키는 방법에 관한 것이다.

배 경 기 술

[0003] 기존 GPU(Graphics Processing Unit)는 그래픽 처리 분야에만 활용되어 왔지만 현재 GPU 기술이 발전함에 따라 중앙 처리 장치(CPU)가 맡았던 응용 프로그램들의 계산에도 활용되고 있다. 이에 따라 더욱 빠른 데이터 처리를

위해 GPU의 성능 개선 연구가 다각도로 진행되고 있다.

[0004] 일반적으로 GPU에서 파일 입출력(Input/Output)이 수행될 때, 디스크에 저장된 데이터는 읽기 명령에 의해 CPU 메모리로 전송된 후, CUDA(Compute Unified Device Architecture)를 통해, GPU 프로세스로부터 읽기 요청된 데이터는 디스크에서 직접 GPU 장치에 포함된 GPU 메모리로 전달되지 않고, CPU 메모리(호스트 메모리)를 경유하여 GPU 메모리로 전달된다. 따라서 GPU에서의 데이터 입출력 동작시에, CPU 부하가 발생할 수 있고, 입출력 성능이 저하될 수 있다.

[0005] 이러한 문제를 해결하기 위해, GPUDirect Storage(GDS) 기술이 개발되었다. GDS는 NVIDIA에서 제공하는 스토리지 가속화 기술로서, NVMe 또는 NVMe-oF와 같은 로컬 혹은 원격 스토리지와 GPU 메모리 간의 직접적인 데이터 경로(Direct I/O)를 지원한다. GDS 기능은 CPU의 바운스 버퍼를 피하여 GPU 메모리로 데이터를 전송하는 네트워크 어댑터 또는 스토리지 근처의 DMA(Direct-Memory Access)엔진을 통해 활성화된다.

[0006] 따라서 GDS 기능이 활성화된 상태에서는, GPU 프로세스로부터 읽기 요청된 데이터가 스토리지에서 직접 GPU 메모리로 전달된다.

[0007] 관련 선행문헌으로 특허 문헌인 대한민국 공개특허 제2019-0098146호, 대한민국 등록특허 제10-1616066호, 비특허 문헌인 "BaM: A Case for Enabling Fine-grain High Throughput GPU-Orchestrated Access to Storage, Zaid Qureshi, Vikram Sharma Mailthody, Isaac Gelado, Seung Won Min, Amna Masood, Jeongmin Park, Jinjun Xiong, CJ Newburn, Dmitri Vainbrand, I-Hsin Chung, Michael Garland, William Dally, Wen-mei Hwu, arXiv:2203.04910"가 있다.

발명의 내용

해결하려는 과제

[0009] 본 발명은 GPU의 데이터 입출력을 가속화시키는 방법을 제공하기 위한 것이다.

[0010] 또한 본 발명은 GDS 기능이 지원되는 환경에서, GPU의 데이터 입출력을 가속화시키는 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0012] 상기한 목적을 달성하기 위한 본 발명의 일 실시예에 따르면, GPU 프로세스로부터 읽기 요청된 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계; 및 상기 파일의 클래스에 따라서, 상기 파일을 CPU 메모리로 전송하거나 또는 GPU 메모리로 전송하는 단계를 포함하는 GPU 데이터 입출력 가속화 방법이 제공된다.

[0013] 또한 상기한 목적을 달성하기 위한 본 발명의 다른 실시예에 따르면, GPU 프로세스로부터 읽기 요청된 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정하는 단계; 및 상기 파일의 클래스에 따라서, 입출력 속도가 서로 다른 저장 장치에 마운트된 제1 및 제2디렉토리 중 하나에 상기 파일을 저장하는 단계를 포함하는 GPU 데이터 입출력 가속화 방법이 제공된다.

발명의 효과

[0015] 본 발명의 일실시예에 따르면, 가속화 클래스로 분류된 GPU 파일을 고속 입출력 저장 장치에 저장함으로써, GPU에서의 데이터 입출력이 더욱 가속화될 수 있다.

[0016] 특히 본 발명의 일실시예에 따르면, GDS 기능이 이용되는 환경에서, 가속화 클래스로 분류된 GPU 파일이 미리 고속 입출력 저장 장치에 저장됨으로써, 보다 빠르게 데이터 입출력이 수행될 수 있다.

도면의 간단한 설명

[0018] 도 1은 본 발명의 일실시예에 따른 GPU 데이터 입출력 가속화 방법을 설명하기 위한 도면이다.

도 2는 본 발명의 일실시예에 따른 클래스 분류 방법을 설명하기 위한 도면이다.

도 3은 본 발명의 일실시예에 따른 파일 입출력 방법을 설명하기 위한 도면이다.

도 4는 본 발명의 일실시예에 따른 GPU 데이터 입출력 가속화 시스템을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0019] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세한 설명에 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.

[0021] 전술된 바와 같이, GPU에서의 데이터 입출력을 가속화하기 위해, 스토리지에 저장된 데이터가 호스트 메모리인 CPU 메모리를 경유하지 않고 직접 GPU 메모리로 전송되도록하는, GDS와 같은 방법이 이용되고 있다.

[0022] GDS 기능은, NVMe SSD와 같은 고속 입출력 저장 장치에 저장된 GPU 데이터를 GPU 메모리로 직접 전송시키기 때문에, GPU에서의 데이터 입출력을 가속화하기 위해서는 GPU 데이터가 고속 입출력 저장 장치에 미리 저장될 필요가 있다. 또한 GDS 기능이 아니더라도, GPU에서의 데이터 입출력을 가속화하기 위해서는 고속 입출력 저장 장치에 GPU 데이터가 저장될 필요가 있다.

[0023] 이에 본 발명은, GPU 프로세스로부터 읽기 요청된 GPU 데이터의 클래스를 분류하고, 클래스에 따라 GPU 데이터를 고속 입출력 저장 장치에 저장하여, 파일 입출력을 처리하는 방법을 제안한다.

[0024] 본 발명의 일실시예는 GPU 데이터의 클래스를 가속화 클래스 및 비가속화 클래스로 분류하고, 가속화 클래스로 분류된 GPU 데이터를 직접 GPU 메모리로 전송한다. 비가속화 클래스로 분류된 GPU 데이터는 CPU 메모리를 경유해 GPU 메모리로 전달된다. 이를 위해 본 발명의 일실시예는 NVMe SSD와 같은 고속 입출력 저장 장치를 캐시화하여, 가속화 클래스로 분류된 GPU 데이터를 고속 입출력 저장 장치에 저장할 수 있다.

[0025] 본 발명의 일실시예에 따른 GPU 데이터 입출력 가속화 방법은 프로세서 및 메모리를 포함하는 컴퓨팅 장치에서 수행될 수 있다.

[0026] 이하에서, 본 발명에 따른 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다.

[0028] 도 1은 본 발명의 일실시예에 따른 GPU 데이터 입출력 가속화 방법을 설명하기 위한 도면이다.

[0029] 도 1을 참조하면, 본 발명의 일실시예에 따른 컴퓨팅 장치는, GPU 프로세스로부터 읽기 요청된 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정(S110)한다. 컴퓨팅 장치는 현재 시스템 환경에서 읽기 요청된 파일에, 가속화 기능을 적용하는 것이 적합한지 여부를 판단하여, 읽기 요청된 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정한다. 가속화 기능을 적용하는 것이 적합하면 파일의 클래스가 가속화 클래스로 결정되고, 가속화 기능을 적용하는 것이 부적합하면, 파일의 클래스가 비가속화 클래스로 결정된다.

[0030] 컴퓨팅 장치는 파일의 클래스를 판단하기 위해 읽기 요청된 파일의 특징 정보와 현재 시스템 정보를 이용할 수 있다. 예컨대, 현재 시스템에서의 자원 사용량이 매우 높으며, 읽기 요청된 파일의 크기가 매우 크다면, 읽기 요청된 파일의 클래스는 비가속화 클래스로 결정될 수 있을 것이다. 그리고 현재 시스템에서의 자원 사용량이 매우 낮으며, 읽기 요청된 파일의 크기가 매우 작다면, 읽기 요청된 파일의 클래스는 가속화 클래스로 결정될 수 있을 것이다. 컴퓨팅 장치는 파일의 클래스를 판단하기 위해, 미리 학습된 분류 모델을 이용할 수 있다.

[0031] 그리고 컴퓨팅 장치는 읽기 요청된 파일의 클래스에 따라서, 파일을 CPU 메모리로 전송하거나 또는 GPU 메모리로 전송(S120)한다. 컴퓨팅 장치는 읽기 요청된 파일의 클래스가 가속화 클래스인 경우, 읽기 요청된 파일을 CPU 메모리를 경유시키지 않고 직접 GPU 메모리로 전송하며, 읽기 요청된 파일의 클래스가 비가속화 클래스인 경우, 읽기 요청된 파일을 CPU 메모리로 전송한다. CPU 메모리로 전송된 파일은, 다시 GPU 메모리로 전송된다.

[0033] 도 2는 본 발명의 일실시예에 따른 클래스 분류 방법을 설명하기 위한 도면이다.

[0034] 도 2를 참조하면 본 발명의 일실시예에 따른 컴퓨팅 장치는 단계 S110에서, 읽기 요청된 파일에 대한 제1특징 정보 및 GPU 프로세스가 실행중인 시스템에 대한 제2특징 정보를 수집(S111)한다. 수집된 정보는, CSV(Comma-Separated Values) 파일에 저장될 수 있다.

[0035] 읽기 요청된 파일에 대한 제1특징 정보는 일실시예로서, 파일의 크기 정보를 포함할 수 있다. 그리고 시스템에 대한 제2특징 정보는 CPU 사용량, GPU 사용량, CPU 메모리 사용량, GPU 메모리 사용량, CPU 메모리의 크기, GPU 메모리의 크기, 시스템의 프레임워크 정보, 입출력 접근 패턴 정보, 입출력 세분성 정보 중 적어도 하나를 포함할 수 있다.

[0036] 여기서, 프레임워크 정보는, 시스템 환경에 대한 정보로서, 현재 시스템 환경이 가상화 환경(docker)인지, 아니

면 가상화가 이용되지 않는 일반 환경(local)인지 또는 클러스터링 환경(ceph)인지를 나타내는 정보이다. 그리고 입출력 접근 패턴(I/O access pattern) 정보는 현재 시스템의 입출력 패턴이 시퀀셜(sequential) 패턴인지, 랜덤(random) 패턴인지를 나타내는 정보이다. 마지막으로 입출력 세분성(I/O granularity) 정보는 데이터의 입출력이, 데이터가 분할되어 수행(fine)되는지, 아니면 분할되지 않고 수행(coarse)되는지를 나타내는 정보이다.

[0037] 그리고 컴퓨팅 장치는 제1 및 제2특징 정보를 미리 학습된 분류 모델에 입력하여, 읽기 요청된 파일의 클래스를 가속화 클래스 및 비가속화 클래스 중 하나로 결정(S112)한다. 분류 모델은 일실시예로서, 인공신경망 모델, 로지스틱 회귀(logistic regression) 모델 또는 랜덤 포레스트(random forest) 모델 등일 수 있다.

[0038] 분류 모델의 학습에 이용되는 훈련 데이터는, 훈련용 제1 및 제2특징 정보와 함께, 훈련용 제1 및 제2특징 정보에 라벨링되는 정답값(ground truth)을 포함할 수 있다. 여기서 정답값은, 훈련용 제1 및 제2특징 정보에 라벨링되는 클래스이다.

[0039] 다양한 훈련용 제1 및 제2특징 정보가 수집되는 환경에서 읽기 요청된 파일에 가속화 기능을 적용하거나 적용하지 않고 데이터 입출력을 수행해봄으로써, 정답값이 획득될 수 있다. 만일 특정 훈련용 제1 및 제2특징 정보가 수집되는 환경에서 읽기 요청된 파일에 가속화 기능을 적용한 경우에서, 그렇지 않은 경우보다 시스템 성능이 우수하다면, 특정 훈련용 제1 및 제2특징 정보에 대한 정답값은 가속화 클래스로 라벨링될 수 있다.

[0040] 단계 S111에서 수집되는 제1 및 제2특징 정보는, 훈련용 제1 및 제2특징 정보에 대응될 수 있다. 즉, 훈련용 제2특징 정보로서, 훈련용 CPU 사용량, 훈련용 GPU 사용량, 훈련용 CPU 메모리 사용량, 훈련용 GPU 메모리 사용량이 이용된 경우, 컴퓨팅 장치는 단계 S111에서 제2특징 정보로서, CPU 사용량, GPU 사용량, CPU 메모리 사용량, GPU 메모리 사용량을 수집할 수 있다.

[0042] 도 3은 본 발명의 일실시예에 따른 파일 입출력 방법을 설명하기 위한 도면이다.

[0043] 도 3을 참조하면 본 발명의 일실시예에 따른 컴퓨팅 장치는 단계 S120에서 파일의 클래스에 따라서, 입출력 속도가 서로 다른 저장 장치에 마운트된 제1 및 제2디렉토리(310, 320) 중 하나에 읽기 요청된 파일을 저장할 수 있다. 가속화 클래스로 분류된 파일은 보다 입출력 속도가 빠른 제1저장 장치에 마운트된 제1디렉토리(310)에 저장되며, 비가속화 클래스로 분류된 파일은, 제1저장 장치보다 입출력 속도가 느린 제2저장 장치에 마운트된 제2디렉토리(320)에 저장될 수 있다. 즉, 제1저장 장치는 GPU 데이터 입출력 가속화를 위한 캐시로 이용되며, 따라서 제1디렉토리(310)는 도 3에서 캐시 디렉토리로 표현되어 있다. 일실시예로서, 제1저장 장치는, NVMe SSD이며, 제2저장 장치는, HDD일 수 있다.

[0044] 제1디렉토리(310)에 저장된 파일은, GPU 메모리(330)와 제1저장 장치 사이의 직접 경로를 통해 GPU 메모리(330)로 전송되며, 제2디렉토리(320)에 저장된 파일은 CPU 메모리(340)로 전송된 후, GPU 메모리(330)로 전달된다. 제1디렉토리(310)에 저장된 제1파일(311)들은 추후 제1파일(311)에 대해 다시 읽기 요청이 발생한 경우, cuFile API와 직접 경로를 통해 GPU 메모리(330)로 전송되며, 제2디렉토리(320)에 저장된 제2파일(321)들은 추후 제2파일(321)에 대해 다시 읽기 요청이 발생한 경우, CPU 메모리(340)로 전송된 후 GPU 메모리(330)로 전달된다.

[0045] 제2디렉토리(320)에는 GPU 프로세스로부터 읽기 요청된 제2파일(321) 뿐만 아니라, CPU 프로세스로부터 읽기 요청된 제3파일(322)들도 저장되며, 제3파일(322)에 대한 읽기 요청이 발생한 경우 제3파일(322)은 CPU 메모리(340)로 전송된다. 제3파일(322)은 GPU 메모리(330)로 전송되지 않는다.

[0046] 제2디렉토리(320)에 제2 및 제3파일(321, 322)들이 저장되는 환경에서 컴퓨팅 장치는 제2 및 제3파일(321, 322)들을 적절한 메모리로 전달하기 위해, 제2디렉토리(320)에 저장된 파일들에 대한 읽기 요청을, GPU 프로세스에 의한 요청 또는 CPU 프로세스에 의한 요청으로 분류한다. 그리고 분류 결과에 따라서, 제2디렉토리(320)에 저장된 파일을 CPU 메모리(340)를 경유시켜 GPU 메모리(330)로 전송하거나 또는 CPU 메모리(340)로 전송할 수 있다.

[0047] 컴퓨팅 장치는 읽기 요청된 파일에 할당된 메모리가 GPU 메모리(330)인지 여부를 판단하여, 제2디렉토리(320)에 저장된 파일들에 대한 읽기 요청을 분류할 수 있다. 읽기 요청된 파일에 할당된 메모리가 GPU 메모리(330)인 경우, 읽기 요청은 GPU 프로세스에 의한 것이며, 읽기 요청된 파일에 할당된 메모리가 CPU 메모리(340)인 경우, 읽기 요청은 CPU 프로세스에 의한 것으로 판단될 수 있다.

[0048] 본 발명의 일실시예에 따르면, 가속화 클래스로 분류된 GPU 파일을 고속 입출력 저장 장치에 저장함으로써, GPU에서의 데이터 입출력이 더욱 가속화될 수 있으며, 특히 GDS 기능이 이용되는 환경에서, 가속화 클래스로 분류된 GPU 파일이 미리 고속 입출력 저장 장치에 저장됨으로써, 보다 빠르게 데이터 입출력이 수행될 수 있다.

[0050] 도 4는 본 발명의 일실시예에 따른 GPU 데이터 입출력 가속화 시스템을 설명하기 위한 도면이다.

[0051] 최초 제2디렉토리(420)에 저장된 파일에 대해 읽기 요청이 발생하면, 제1 및 제2특징 정보가 데이터 수집 모듈(450)에 의해 수집된다. 읽기 요청은 GPU 프로세스에 의하거나 CPU 프로세스에 의한 것일 수 있다. 이후, 미리 학습된 분류 모델(460)에 의해 읽기 요청된 파일의 클래스가 가속화 클래스(GDS) 및 비가속화 클래스(Non GDS) 중 하나로 결정된다.

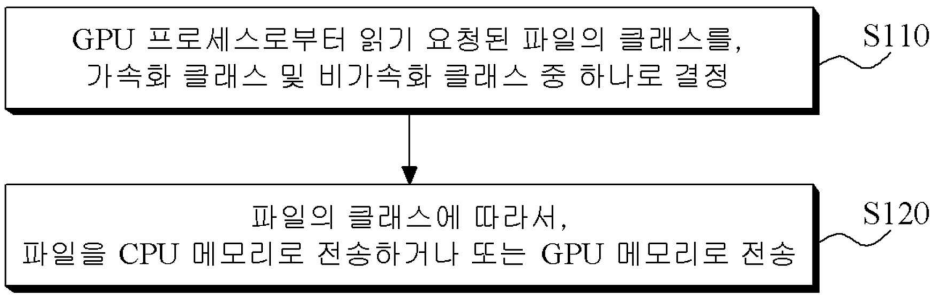
[0052] 읽기 요청된 파일의 클래스가 가속화 클래스인 경우 해당 파일은 제1디렉토리(410)에 저장되고, NVMe SSD와 GPU 메모리 사이의 직접 경로를 통해 GPU 메모리(430)로 전송된다. GPU 메모리(430)로 전송된 파일은 GPU에서 처리된다. 읽기 요청된 파일의 클래스가 비가속화 클래스인 경우 해당 파일은 그대로 제2디렉토리(420)에 저장되고 CPU 메모리(440)로 전송된다. CPU 메모리(440)로 전송된 파일은 CPU에서 처리되거나, GPU 메모리(430)로 전송되어 GPU에서 처리된다.

[0054] 앞서 설명한 기술적 내용들은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예들을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 하드웨어 장치는 실시예들의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

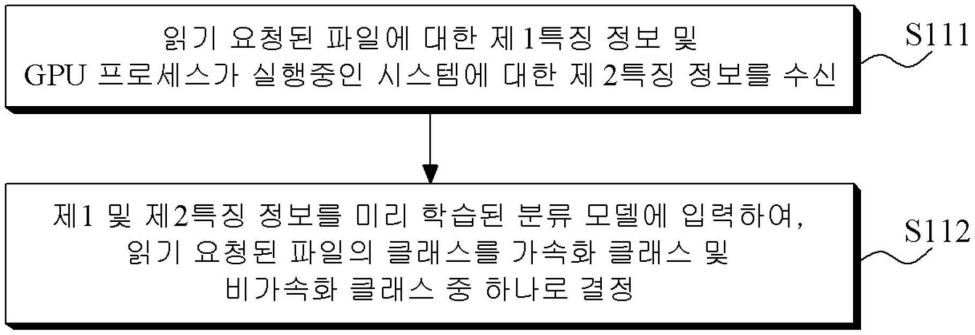
[0056] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다. 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고 할 것이다.

도면

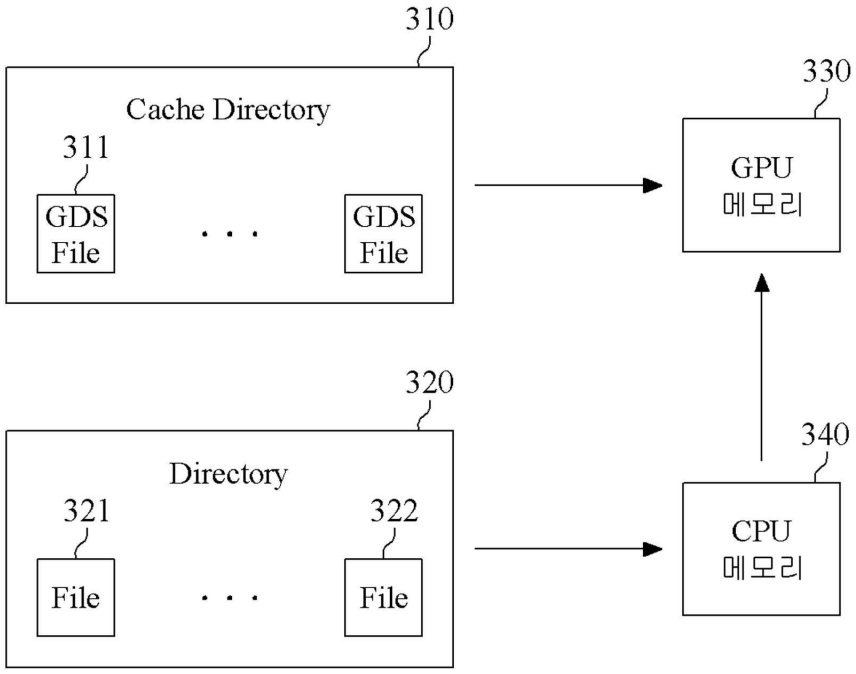
도면1



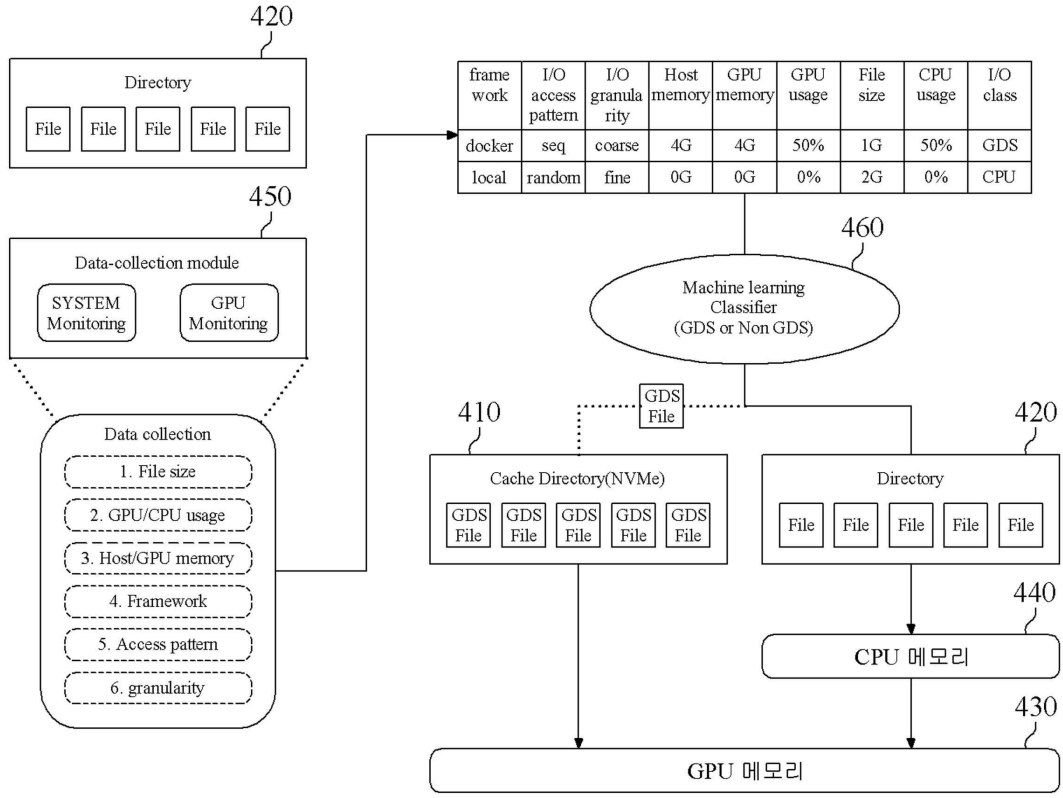
도면2


읽기 요청된 파일에 대한 제1특징 정보 및
GPU 프로세스가 실행중인 시스템에 대한 제2특징 정보를 수신
S111
제1 및 제2특징 정보를 미리 학습된 분류 모델에 입력하여,
읽기 요청된 파일의 클래스를 가속화 클래스 및
비가속화 클래스 중 하나로 결정
S112

도면3


310
Cache Directory
311
GDS File
GDS File
330
GPU 메모리
320
Directory
321
322
File
File
340
CPU 메모리

도면4


420
Directory
File
File
File
File
File
450
Data-collection module
SYSTEM Monitoring
GPU Monitoring
Data collection
1. File size
2. GPU/CPU usage
3. Host/GPU memory
4. Framework
5. Access pattern
6. granularity
frame work
I/O access pattern
I/O granularity
Host memory
GPU memory
GPU usage
File size
CPU usage
I/O class
docker
seq
coarse
4G
4G
50%
1G
50%
GDS
local
random
fine
0G
0G
0%
2G
0%
CPU
460
Machine learning Classifier (GDS or Non GDS)
GDS File
410
Cache Directory(NVMe)
GDS File
GDS File
GDS File
GDS File
GDS File
420
Directory
File
File
File
File
File
440
CPU 메모리
430
GPU 메모리