



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0078331
(43) 공개일자 2020년07월01일

- (51) 국제특허분류(Int. Cl.)
G06F 9/50 (2018.01) G06F 13/40 (2006.01)
G06F 9/30 (2018.01)
- (52) CPC특허분류
G06F 9/5027 (2013.01)
G06F 13/4027 (2013.01)
- (21) 출원번호 10-2019-0159399
- (22) 출원일자 2019년12월03일
심사청구일자 없음
- (30) 우선권주장
62/784,275 2018년12월21일 미국(US)
16/543,264 2019년08월16일 미국(US)

- (71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
- (72) 발명자
카차르, 람다스 피.
미국 캘리포니아주 94588 플레젠턴 페어브룩 코트 7798
피스처, 스테판 쥐.
미국 캘리포니아주 95126 산호세 로도스 코트 109
핀토, 오스카 피.
미국 캘리포니아주 95132 산호세 베타니 애비뉴 1774
- (74) 대리인
특허법인 고려

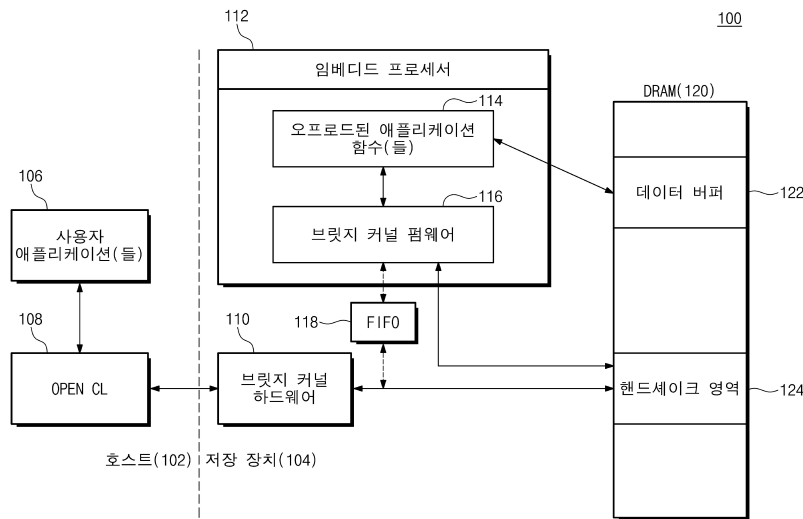
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 장치로 애플리케이션 기능들을 오프로드하는 시스템 및 방법

(57) 요약

시스템은 호스트 장치, 임베디드 프로세서를 포함하는 저장 장치, 및 브릿지 커널 하드웨어 및 브릿지 커널 펌웨어를 포함하는 브릿지 커널 장치를 포함하는 시스템을 포함하되, 브릿지 커널 장치는 호스트 장치로부터 복수의 인자를 수신하고, 데이터 처리를 위해 임베디드 프로세서로 복수의 인자를 전송하도록 구성된다.

대표도



(52) CPC특허분류

G06F 9/30098 (2013.01)

G06F 9/5016 (2013.01)

G06F 2213/0026 (2013.01)

명세서

청구범위

청구항 1

호스트 장치;

임베디드 프로세서를 포함하는 저장 장치; 및

브릿지 커널 하드웨어 및 브릿지 커널 펌웨어를 포함하는 브릿지 커널 장치를 포함하되,

상기 브릿지 커널 장치는 상기 호스트 장치로부터 복수의 인자를 수신하고, 데이터 처리를 위해 상기 임베디드 프로세서로 상기 복수의 인자를 전송하도록 구성되는 시스템.

청구항 2

제 1 항에 있어서,

상기 브릿지 커널 장치의 구현은 상기 호스트 장치에 투명하고,

상기 브릿지 커널 장치는 상기 호스트 장치와 상기 브릿지 커널 장치간의 전송 메커니즘에 무관하고,

상기 전송 메커니즘은 PCIe(Peripheral Component Interconnect express) 또는 이더넷 연결 중 하나인 시스템.

청구항 3

제 1 항에 있어서,

상기 저장 장치는 SSD(Solid State Device)이고,

상기 임베디드 프로세서는 FPGA(Field Programmable Gate Array) 프로세서 또는 SSD 컨트롤러 또는 이산 보조 프로세서이고,

상기 임베디드 프로세서는 상기 임베디드 프로세서 상에서 실행되도록 구성되는 오프로드된 애플리케이션 기능들을 포함하고, 상기 브릿지 커널 펌웨어를 더 포함하고,

상기 브릿지 커널 장치는 상기 호스트 장치로부터 수신된 복수의 인자를 상기 오프로드된 애플리케이션 기능들로 전송하고, 상기 호스트 장치의 호스트 레지스터 인터페이스로부터 수신된 상기 복수의 인자를 폐치하는 시스템.

청구항 4

제 3 항에 있어서,

상기 스토리지 장치는 FIFO(First In, First Out) 레지스터 및 DRAM(Dynamic Random-Access Memory)을 포함하고,

상기 DRAM은 상기 호스트 장치로부터 수신된 상기 복수의 인자를 저장하기 위해 복수의 데이터 버퍼 및 핸드셰이크 영역을 포함하고,

상기 FIFO 레지스터 및 상기 DRAM은 상기 브릿지 커널 하드웨어와 통신 가능하도록 연결되는 시스템.

청구항 5

제 4 항에 있어서,

상기 FIFO 레지스터는 데이터 처리를 위해 상기 오프로드된 애플리케이션 기능들을 포함하는 상기 임베디드 프로세서로 상기 복수의 인자를 통과시키도록 구성되는 시스템.

청구항 6

제 4 항에 있어서,

상기 브릿지 커널 하드웨어는 상기 호스트 장치로부터 수신된 상기 복수의 인자를 상기 DRAM의 상기 핸드셰이크 영역에 일시적으로 저장하도록 구성되는 시스템.

청구항 7

제 4 항에 있어서,

상기 브릿지 커널 장치는 상기 호스트 장치의 고객 애플리케이션 모듈 상에서 실행되는 호스트 애플리케이션에 대해 프록시로써 동작하도록 구성되고,

상기 호스트 애플리케이션은 상기 오프로드된 애플리케이션 기능들이 어디서 또는 어떻게 실행되는지와 무관한 시스템.

청구항 8

제 4 항에 있어서,

상기 호스트 장치는 고객 애플리케이션 모듈 및 OpenCL(Open Computing Language) 모듈을 포함하고,

상기 호스트 장치는 상기 저장 장치를 이용하여 PCIe(Peripheral Component Internet express) 또는 이더넷 연결을 설정하도록 구성되는 시스템.

청구항 9

제 8 항에 있어서,

상기 브릿지 커널 하드웨어는 복수의 하드웨어 커널을 포함하고,

상기 브릿지 커널 펌웨어는 상기 임베디드 프로세서 상에서 실행되도록 구성되는 복수의 소프트웨어 커널을 포함하고,

상기 호스트 장치의 상기 고객 애플리케이션 모듈 상에서 실행되는 호스트 애플리케이션은 상기 OpenCL 모듈을 통해 상기 복수의 소프트웨어 커널 및 상기 복수의 하드웨어 커널 중 적어도 하나와 접속하도록 구성되는 시스템.

청구항 10

제 9 항에 있어서,

상기 브릿지 커널 장치는 비용 기능에 기초하여 데이터 처리를 위해 상기 하드웨어 커널 및 상기 복수의 소프트웨어 커널 중에서 하나 이상의 커널을 선택하도록 구성되는 시스템.

청구항 11

제 10 항에 있어서,

상기 브릿지 커널 장치는 상기 복수의 인자 및 상기 호스트 장치로부터 수신된 스토리지 또는 네트워킹 파라미터를 이용하여, 상기 비용 기능에 기초하여 데이터 처리를 위해 상기 하나 이상의 커널을 선택하도록 구성되고, 상기 호스트 장치로부터 수신된 상기 스토리지 또는 상기 네트워킹 파라미터는 SQID(Submission Queue Identifier), CQID(Completion Queue Identifier), 스트림 ID, 호스트 ID, LBA(Logical Block Address) 범위, NSID(Network Service ID), MAC(Media Access Control) ID, TCP(Transmission Control Protocol)/IP(Internet Protocol) 필드, 애플리케이션 식별자, 또는 상기 호스트와 관련된 시간 및 날짜 중 적어도 하나를 포함하는 시스템.

청구항 12

제 11 항에 있어서,

상기 브릿지 커널 장치는 상기 FIFO 레지스터를 이용하여 데이터 처리를 위해 상기 복수의 하드웨어 커널 및 상기 복수의 소프트웨어 커널 중에서 하나 이상의 커널을 선택하도록 구성되는 시스템.

청구항 13

제 12 항에 있어서,

상기 브릿지 커널 펌웨어는 상기 DRAM의 상기 핸드셰이크 영역 내의 상기 복수의 인자를 폐치하도록 구성되고, 상기 호스트 장치로부터 수신된 상기 복수의 인자를 갖는 상기 오프로드된 애플리케이션 기능들을 호출하도록 구성되는 시스템.

청구항 14

제 13 항에 있어서,

상기 오프로드된 애플리케이션 기능들은 데이터 처리를 위해 상기 복수의 인자를 사용하도록 구성되는 시스템.

청구항 15

저장 장치 내의 브릿지 커널 장치에 의해, 상기 저장 장치와 연결된 호스트 장치로부터 트리거를 수신하는 단계;

상기 브릿지 커널 장치에 의해, 상기 호스트 장치로부터 수신된 복수의 인자를 폐치하는 단계;

상기 브릿지 커널 장치에 의해, 상기 저장 장치의 DRAM 내에 상기 복수의 인자를 저장하는 단계;

상기 브릿지 커널 장치에 의해, 상기 DRAM 내에 제 1 준비 플래그를 설정하는 단계;

상기 브릿지 커널 장치에 의해, 상기 복수의 인자에 기초하여 상기 저장 장치의 임베디드 프로세서 내의 상기 브릿지 커널 장치의 펌웨어 내의 오프로드된 기능들의 처리가 완료되었는지를 결정하는 상기 브릿지 커널 장치에 기초하여, 상기 DRAM 내의 완료 플래그를 폴링하는 단계; 및

상기 브릿지 커널 장치에 의해, 상기 브릿지 커널 장치가 상기 복수의 인자를 폐치하도록 구성되는 호스트 레지스터 인터페이스 내에 제 2준비 플래그가 설정되는 단계를 포함하되,

상기 복수의 인자는 상기 DRAM의 핸드셰이크 영역 내에 저장되고,

상기 제 1 준비 플래그는 상기 DRAM의 상기 핸드셰이크 영역 내에 설정되고,

상기 완료 플래그는 상기 DRAM의 상기 핸드셰이크 영역 내에 폴링되는 방법.

청구항 16

제 15 항에 있어서,

상기 브릿지 커널 장치에 의해, 상기 DRAM의 상기 핸드셰이크 영역 내의 상기 완료 플래그를 감지하는 것에 기초하여, 상기 DRAM의 상기 핸드셰이크 영역으로부터 에러 또는 상태를 판독하는 단계; 및

상기 브릿지 커널 장치에 의해, 상기 호스트 레지스터 인터페이스로 상기 에러 또는 상기 상태를 업데이트 하는 단계를 더 포함하는 방법.

청구항 17

제 15 항에 있어서,

상기 브릿지 커널 장치의 상기 펌웨어에 의해, 상기 DRAM 내에 제 1 준비 플래그를 폴링하는 단계;

상기 브릿지 커널 장치의 상기 펌웨어에 의해, 상기 DRAM의 핸드셰이크 영역 내에 복수의 인자를 폐치하는 단계;

상기 브릿지 커널 장치의 상기 펌웨어에 의해, 데이터 처리를 위해 상기 복수의 인자를 갖는 상기 스토리지 장치의 상기 임베디드 프로세서 내의 오프로드된 애플리케이션 기능들을 호출하는 단계;

상기 오프로드된 애플리케이션 기능들에 의해, 상기 데이터 처리의 완료에 기초하여 상기 브릿지 커널 장치의 상기 펌웨어로 상기 브릿지 커널 장치의 상기 펌웨어에 의해 호출된 상기 오프로드된 애플리케이션 기능들을 반환하는 단계; 및

상기 브릿지 커널 장치의 상기 펌웨어에 의해, 상기 DRAM의 상기 핸드셰이크 영역 내에 완료 플래그를 설정하는 단계를 더 포함하는 방법.

청구항 18

임베디드 프로세서를 포함하는 저장 장치; 및

호스트 장치로부터 복수의 인자를 수신하고, 데이터 처리를 위해 상기 복수의 인자를 상기 임베디드 프로세서로 전송하는 브릿지 커널 장치를 포함하는 시스템.

청구항 19

제 18 항에 있어서,

상기 브릿지 커널 장치는 브릿지 커널 하드웨어 및 브릿지 커널 펌웨어를 포함하고,

상기 저장 장치는 SSD(Solid State Drive)이고,

상기 임베디드 프로세서는 FPGA(Field Programmable Gate Array) 프로세서 또는 SSD 컨트롤러 또는 이산 보조 프로세서이고,

상기 임베디드 프로세서는 오프로드된 애플리케이션 기능들 및 상기 브릿지 커널 펌웨어를 포함하고;

상기 브릿지 커널 장치는 상기 호스트 장치로부터 수신된 상기 복수의 인자를 상기 임베디드 프로세서 상에서 실행되도록 구성되는 상기 오프로드된 애플리케이션 기능들로 전송하고, 상기 호스트 장치로부터 수신된 복수의 인자를 상기 호스트 레지스터 인터페이스로부터 폐치하는 시스템.

청구항 20

제 19 항에 있어서,

상기 저장 장치는 FIFO(First In First Out) 레지스터 및 DRAM(Dynamic Random Access Memory)를 포함하고,

상기 DRAM은 복수의 데이터 버퍼 및 핸드셰이크 영역을 포함하고,

상기 FIFO 레지스터 및 상기 DRAM은 상기 브릿지 커널 하드웨어와 통신 가능하도록 연결되고,

상기 브릿지 커널 하드웨어는 복수의 하드웨어 커널 및 복수의 소프트웨어 커널을 포함하고,

상기 복수의 소프트웨어 커널은 상기 임베디드 프로세서에서 실행되도록 구성되고,

상기 브릿지 커널 장치는 상기 복수의 인자 및 상기 호스트 장치로부터 수신된 스토리지 또는 네트워킹 매개 변수들을 이용하여, 비용 기능에 기초하여 데이터 처리를 위해 상기 복수의 하드웨어 커널 및 상기 복수의 소프트웨어 커널 중에서 하나 이상의 커널을 선택하도록 구성되고,

상기 호스트 장치로부터 수신된 상기 스토리지 또는 상기 네트워킹 파라미터는 SQID(Submission Queue Identifier), CQID(Completion Queue Identifier), 스트림 ID, 호스트 ID, LBA(Logical Block Address) 범위, NSID(Network Service ID), MAC(Media Access Control) ID, TCP(Transmission Control Protocol)/IP(Internet Protocol) 필드, 애플리케이션 식별자, 또는 상기 호스트와 관련된 시간 및 날짜 중 하나 이상을 포함하고,

상기 브릿지 커널 장치는 상기 FIFO 레지스터를 사용하여 데이터 처리를 위해 상기 복수의 하드웨어 커널 및 상기 복수의 소프트웨어 커널 중에서 상기 하나 이상의 커널을 선택하도록 구성되는 시스템.

발명의 설명

기술 분야

본 발명은 오프로드하는 시스템 및 방법에 관한 것으로, 좀 더 상세하게는 장치로 애플리케이션 기능들을 분담(offload)하는 시스템 및 방법에 관한 것이다.

배경 기술

[0001]

[0002] 최신 정보 기술(IT) 인프라에서, 다양한 장치 및 프로세스에 의해 비교적 많은 양의 데이터가 생성될 수 있다. 이러한 데이터 생성기의 몇몇 예들은 스마트 장치들(예로서, 아이폰(iPhone), 아이패드(iPad) 등), 자율 주행 차량들, 소셜 네트워크 및 사물 인터넷(IOT) 장치들을 포함한다. 수집된 데이터를 분석하고 사용하여 높은 애플리케이션 효율성과 생산성을 달성하기 위해 인공 지능(Artificial Intelligence, AI) 및 기계 학습(Machine Learning, ML) 알고리즘이 개발되고 있다. 최근 IT 인프라의 발전에 비추어 데이터를 처리하고 분석하는 것이 바람직할 수 있다.

[0003] '발명의 배경이 되는 기술' 부분의 상기 정보는 단지 기술의 배경에 대한 이해를 증진시키기 위한 것일 뿐이므로, 종래 기술의 존재 또는 관련성을 인정하는 것으로 해석되어서는 안된다.

발명의 내용

해결하려는 과제

[0004] 본 발명은 상술된 기술적 과제를 해결하기 위한 것으로서, 본 발명의 목적은 장치로 애플리케이션 기능들을 오프로드하는 시스템 및 방법을 제공하는 데 있다.

과제의 해결 수단

[0005] 본 개시의 실시 예의 일면은 장치로 애플리케이션 기능들을 오프로드하기 위한 시스템 및 방법과 연관된다. 본 개시의 몇몇 실시 예에 따르면, 시스템은 호스트 장치, 임베디드 프로세서를 포함하는 저장 장치, 및 브릿지 커널 하드웨어 및 브릿지 커널 펌웨어를 포함하는 브릿지 커널 장치를 포함하는 시스템을 포함하되, 브릿지 커널 장치는 호스트 장치로부터 복수의 인자를 수신하고, 데이터 처리를 위해 임베디드 프로세서로 복수의 인자를 전송하도록 구성된다.

[0006] 몇몇 실시 예에서, 브릿지 커널 장치의 구현은 호스트 장치에 투명하고, 브릿지 커널 장치는 호스트 장치와 브릿지 커널 장치간의 전송 메커니즘에 무관하고, 전송 메커니즘은 PCIe(Peripheral Component Interconnect express) 또는 이더넷 연결 중 하나이다. 몇몇 실시 예에서, 저장 장치는 SSD(Solid State Device)이고, 임베디드 프로세서는 FPGA(Field Programmable Gate Array) 프로세서 또는 SSD 컨트롤러 또는 이산 보조 프로세서이고, 임베디드 프로세서는 임베디드 프로세서 상에서 실행되도록 구성되는 오프로드된 애플리케이션 기능들을 포함하고, 브릿지 커널 펌웨어를 더 포함하고, 브릿지 커널 장치는 호스트 장치로부터 수신된 복수의 인자를 오프로드된 애플리케이션 기능들로 전송하고, 호스트 장치의 호스트 레지스터 인터페이스로부터 수신된 복수의 인자를 폐지한다.

[0007] 몇몇 실시 예에서, 스토리지 장치는 FIFO(First In, First Out) 레지스터 및 DRAM(Dynamic Random-Access Memory)을 포함하고, DRAM은 호스트 장치로부터 수신된 복수의 인자를 저장하기 위해 복수의 데이터 버퍼 및 핸드셰이크 영역을 포함하고, FIFO 레지스터 및 DRAM은 브릿지 커널 하드웨어와 통신 가능하도록 연결된다. 몇몇 실시 예에서, FIFO 레지스터는 데이터 처리를 위해 오프로드된 애플리케이션 기능들을 포함하는 임베디드 프로세서로 복수의 인자를 통과시키도록 구성된다. 몇몇 실시 예에서, 브릿지 커널 하드웨어는 호스트 장치로부터 수신된 복수의 인자를 DRAM의 핸드셰이크 영역에 일시적으로 저장하도록 구성된다. 몇몇 실시 예에서, 브릿지 커널 장치는 호스트 장치의 고객 애플리케이션 모듈 상에서 실행되는 호스트 애플리케이션에 대해 프록시로서 동작하도록 구성되고, 호스트 애플리케이션은 오프로드된 애플리케이션 기능들이 어디서 또는 어떻게 실행되는지와 무관하다.

[0008] 몇몇 실시 예에서, 호스트 장치는 고객 애플리케이션 모듈 및 OpenCL(Open Computing Language) 모듈을 포함하고, 호스트 장치는 저장 장치를 이용하여 PCIe(Peripheral Component Internet express) 또는 이더넷 연결을 설정하도록 구성된다. 몇몇 실시 예에서, 브릿지 커널 하드웨어는 복수의 하드웨어 커널을 포함하고, 브릿지 커널 펌웨어는 임베디드 프로세서 상에서 실행되도록 구성되는 복수의 소프트웨어 커널을 포함하고, 호스트 장치의 고객 애플리케이션 모듈 상에서 실행되는 호스트 애플리케이션은 OpenCL 모듈을 통해 복수의 소프트웨어 커널 및 복수의 하드웨어 커널 중 적어도 하나와 접속하도록 구성된다. 몇몇 실시 예에서, 브릿지 커널 장치는 비용 기능에 기초하여 데이터 처리를 위해 하드웨어 커널 및 복수의 소프트웨어 커널 중에서 하나 이상의 커널을 선택하도록 구성된다.

[0009] 몇몇 실시 예에서, 브릿지 커널 장치는 복수의 인자 및 호스트 장치로부터 수신된 스토리지 또는 네트워킹 파라미터를 이용하여, 비용 기능에 기초하여 데이터 처리를 위해 하나 이상의 커널을 선택하도록 구성되고, 호스트

장치로부터 수신된 스토리지 또는 네트워킹 파라미터는 SQID(Submission Queue Identifier), CQID(Completion Queue Identifier), 스트림 ID, 호스트 ID, LBA(Logical Block Address) 범위, NSID(Network Service ID), MAC(Media Access Control) ID, TCP(Transmission Control Protocol)/IP(Internet Protocol) 필드, 애플리케이션 식별자, 또는 호스트와 관련된 시간 및 날짜 중 적어도 하나를 포함한다. 몇몇 실시 예에서, 브릿지 커널 장치는 FIFO 레지스터를 이용하여 데이터 처리를 위해 복수의 하드웨어 커널 및 복수의 소프트웨어 커널 중에서 하나 이상의 커널을 선택하도록 구성된다. 몇몇 실시 예에서, 브릿지 커널 펌웨어는 DRAM의 핸드셰이크 영역 내의 복수의 인자를 폐치하도록 구성되고, 호스트 장치로부터 수신된 복수의 인자를 갖는 오프로드된 애플리케이션 기능들을 호출하도록 구성된다. 몇몇 실시 예에서, 오프로드된 애플리케이션 기능들은 데이터 처리를 위해 복수의 인자를 사용하도록 구성된다.

[0010] 여기에서 개시되는 몇몇 실시 예에 따르면, 방법은 저장 장치 내의 브릿지 커널 장치에 의해, 저장 장치와 연결된 호스트 장치로부터 트리거를 수신하는 단계, 브릿지 커널 장치에 의해, 호스트 장치로부터 수신된 복수의 인자를 폐치하는 단계, 브릿지 커널 장치에 의해, 저장 장치의 DRAM 내에 복수의 인자를 저장하는 단계, 브릿지 커널 장치에 의해, DRAM 내에 제 1 준비 플래그를 설정하는 단계, 브릿지 커널 장치에 의해, 복수의 인자에 기초하여 저장 장치의 임베디드 프로세서 내의 브릿지 커널 장치의 펌웨어 내의 오프로드된 기능의 처리가 완료되었는지를 결정하는 브릿지 커널 장치에 기초하여, DRAM 내의 완료 플래그를 폴링하는 단계, 및 브릿지 커널 장치에 의해, 호스트 레지스터 인터페이스 내에 제 2준비 플래그가 설정되는 단계를 포함하되, 브릿지 커널 장치는 호스트 레지스터 인터페이스로부터 복수의 인자를 폐치하도록 구성되고, 복수의 인자는 DRAM의 핸드셰이크 영역 내에 저장되고, 제 1 준비 플래그는 DRAM의 핸드셰이크 영역 내에 설정되고, 완료 플래그는 DRAM의 핸드셰이크 영역 내에 폴링된다.

[0011] 몇몇 실시 예에서, 방법은 브릿지 커널 장치에 의해, DRAM의 핸드셰이크 영역 내의 완료 플래그를 감지하는 것에 기초하여, DRAM의 핸드셰이크 영역으로부터 에러 또는 상태를 판독하는 단계, 및 브릿지 커널 장치에 의해, 호스트 레지스터 인터페이스로 에러 또는 상태를 업데이트 하는 단계를 더 포함한다. 몇몇 실시 예에서, 방법은 브릿지 커널 장치의 펌웨어에 의해, DRAM 내에 제 1 준비 플래그를 폴링하는 단계, 브릿지 커널 장치의 펌웨어에 의해, DRAM의 핸드셰이크 영역 내에 복수의 인자를 폐치하는 단계, 브릿지 커널 장치의 펌웨어에 의해, 데이터 처리를 위해 복수의 인자를 갖는 스토리지 장치의 임베디드 프로세서 내의 오프로드된 애플리케이션 기능들을 호출하는 단계, 오프로드된 애플리케이션 기능들에 의해, 데이터 처리의 완료에 기초하여 브릿지 커널 장치의 펌웨어로 브릿지 커널 장치의 펌웨어에 의해 호출된 오프로드된 애플리케이션 기능들을 반환하는 단계, 및 브릿지 커널 장치의 펌웨어에 의해, DRAM의 핸드셰이크 영역 내에 완료 플래그를 설정하는 단계를 포함하는 방법을 포함한다.

[0012] 본 개시의 몇몇 실시 예에 따르면, 시스템은 임베디드 프로세서를 포함하는 저장 장치, 및 호스트 장치로부터 복수의 인자를 수신하고, 데이터 처리를 위해 복수의 인자를 임베디드 프로세서로 전송하는 브릿지 커널 장치를 포함한다. 몇몇 실시예에서, 브릿지 커널 장치는 브릿지 커널 하드웨어 및 브릿지 커널 펌웨어를 포함하고, 저장 장치는 SSD(Solid State Drive)이고, 임베디드 프로세서는 FPGA(Field Programmable Gate Array) 프로세서 또는 SSD 컨트롤러 또는 이산 보조 프로세서이고, 임베디드 프로세서는 오프로드된 애플리케이션 기능들 및 브릿지 커널 펌웨어를 포함하고, 브릿지 커널 장치는 호스트 장치로부터 수신된 복수의 인자를 임베디드 프로세서 상에서 실행되도록 구성되는 오프로드된 애플리케이션 기능들로 전송하고, 호스트 장치로부터 수신된 복수의 인자를 호스트 레지스터 인터페이스로부터 폐치한다.

[0013] 몇몇 실시 예에서, 저장 장치는 FIFO(First In First Out) 레지스터 및 DRAM(Dynamic Random Access Memory)를 포함하고, DRAM은 복수의 데이터 버퍼 및 핸드셰이크 영역을 포함하고, FIFO 레지스터 및 DRAM은 브릿지 커널 하드웨어와 통신 가능하도록 연결되고, 브릿지 커널 하드웨어는 복수의 하드웨어 커널 및 복수의 소프트웨어 커널을 포함하고, 복수의 소프트웨어 커널은 임베디드 프로세서에서 실행되도록 구성되고, 브릿지 커널 장치는 복수의 인자 및 호스트 장치로부터 수신된 스토리지 또는 네트워킹 매개 변수들을 이용하여, 비용 기능에 기초하여 데이터 처리를 위해 복수의 하드웨어 커널 및 복수의 소프트웨어 커널 중에서 하나 이상의 커널을 선택하도록 구성되고, 호스트 장치로부터 수신된 스토리지 또는 네트워킹 파라미터는 SQID(Submission Queue Identifier), CQID(Completion Queue Identifier), 스트림 ID, 호스트 ID, LBA(Logical Block Address) 범위, NSID(Network Service ID), MAC(Media Access Control) ID, TCP(Transmission Control Protocol)/IP(Internet Protocol) 필드, 애플리케이션 식별자, 또는 호스트와 관련된 시간 및 날짜 중 하나 이상을 포함하고, 브릿지 커널 장치는 FIFO 레지스터를 사용하여 데이터 처리를 위해 복수의 하드웨어 커널 및 복수의 소프트웨어 커널 중에서 하나 이상의 커널을 선택하도록 구성된다.

발명의 효과

[0014] 본 발명은 소프트웨어 기반 오프로드된 기능을 이용하여 호스트에서 이미 해당 기능을 수행하는 소프트웨어 코드를 재사용 할 수 있다. 또한, 본 발명은(기능을) 호스트 소프트웨어 스택과 오프로드된 커널들간의 인터페이스를 소프트웨어 커널들 및 다른 형태의 커널 구현과 동일하게 유지할 수 있다. 따라서, 초기의 빠른 오프로드가 소프트웨어 커널 형태에서 수행될 수 있고, 나중에 오프로드된 기능은 전체 호스트 소프트웨어 스택 및/또는 시스템 아키텍처를 크게 변경하거나 중단하지 않고 하드웨어에서 구현될 수 있다. 빠른 오프로드 및 실행을 통해, 더 높은 성능의 커널들이 개발되는 동안 다른 시스템 활동(예로서, 시스템 통합, 검증, 특성화 등)이 진행될 수 있다.

[0015] 즉, 본 발명은 효율적이고 비용 효율적인 데이터 처리를 위해, 애플리케이션 기능들을 임베디드 프로세서를 포함하는 SSD와 같은 컴퓨터의(연산) 저장 장치로 오프로드할 수 있다.

도면의 간단한 설명

- [0016] 도 1a는 브릿지 커널을 포함하는 시스템의 예시적인 블록도를 도시한다.
- 도 1b는 브릿지 커널에 의해 상이한 커널들로의 비용 기반 라우팅 프로세스를 도시한다.
- 도 2a 내지 2b는 오프로드된 커널과 접속하는 호스트 애플리케이션의 예시적인 흐름을 도시한다.
- 도 3은 브릿지 커널에 의해 호스트에 제시된 레지스터 인터페이스의 예를 도시한다.
- 도 4는 도 1에 도시된 것처럼 핸드셰이크 영역의 레이아웃의 예를 도시한다.
- 도 5는 브릿지 커널 하드웨어를 상태 기계로서 사용하는 방법을 도시한다.
- 도 6은 브릿지 커널 펌웨어 동작을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0017] 첨부된 도면들과 관련하여 아래에 설명되는 상세한 설명은 본 발명에 따라 제공된 장치에 애플리케이션 기능들을 오프로드(off-load)하기 위한 시스템 및 방법의 일부 예시적인 실시 예들의 설명으로서 의도된 것이며, 본 개시가 구성되거나 이용될 수 있는 유일한 형태를 나타내기 위해 의도된 것은 아니다. 본 설명은 도시된 실시 예들과 관련하여 본 개시의 특징을 설명한다. 그러나, 동일하거나 동등한 기능 및 구조는 또한 본 개시의 범위 내에 포함되도록 의도된 상이한 실시 예에 의해 달성될 수 있음을 이해해야 한다. 본 설명의 다른 곳에 표시된 바와 같이, 유사한 요소 번호는 유사한 요소 또는 특징을 나타내도록 의도된다.

[0018] 몇몇 시스템 아키텍처에서, 데이터는 영구적인 저장소로부터 고성능 네트워크를 사용하여 저장소에 연결될 수 있는 비교적 고성능인 서버로 페치될 수 있다. 처리 및 분석을 위해 중앙 처리 장치(CPU)에 대량의 원시 데이터와 같은 이동은 에너지 소비 및 배치된 네트워크 및 컴퓨팅(연산) 자원의 양의 관점에서 비용이 많이 들게 될 수 있다. 이러한, 많은 양의 원시 데이터의 이동은 네트워크 대역폭, CPU 주기 및 CPU 메모리와 같은 자원의 부담을 증가시킬 수 있다. 처리를 위해 서버에 많은 양의 원시 데이터를 이동하는 것은 서버에서 실행되는 애플리케이션에 의해 경험되는 지연(레이턴시)을 또한 증가시킬 수 있다.

[0019] 일부 상황에서, 애플리케이션은 처리가 완료되고 결정이 이루어지기 전에 데이터가 우선 서버에 페치될 때까지 기다려야 할 수 있다. 이러한 추가 자원 요구 사항으로 인해 높은 자본 지출과 운영 지출 비용이 발생할 수 있다. 따라서, 저장 장치(예로서, SSD(solid state device)) 내에서 원시 데이터를 처리하는 것(가능한 한 많이)은 많은 데이터 분석 사용 사례(예로서, 증가하는 원시 데이터 양을 통한 수익 창출을 지향하는 사용 사례)에 대한 비용 효율적인 해결 방안을 나타낼 수 있다. 또한, 데이터 이동을 줄임으로서, 애플리케이션을 실행하는 시스템의 응답 대기 시간을 향상시킬 수 있다.

[0020] 게다가, 데이터 분석 작업을 이용하는 시스템은 종종 비교적 많은 양의 데이터를 읽고, 그것을 처리하고, 필터링 및 다른 데이터 감소 동작을 통해 그것을 감소시킬 수 있다. 이러한 작업은 축소된 결과를 이동하는 것만으로, 장치(예로서, SSD) 내에서 더 높은 가용 대역폭을 활용하고 SSD와 CPU 사이의(예로서, RDAS(Remote Direct Attached Storage)의 경우 네트워크 패브릭 인터페이스 또는 PCIe(Peripheral Component Interconnect Express) 인터페이스) 제한된 대역폭을 보존함으로써, 이러한 작업 및 기타 유사한 작업이 SSD 기반 서비스로 구현될 수 있다. 따라서, 저장 장치(예로서, SSD)에 더 가까이 또는 내부에서 데이터를 처리하는 기술은 효율적

이고 비용 효율적인 데이터 처리를 위해 바람직하다.

- [0021] 본 개시의 일부 실시 예는 애플리케이션 기능을 컴퓨터의 저장 장치(예로서, 임베디드 프로세서를 포함하는 SSD)로 오프로드하기 위한 방법 및 시스템을 포함하며, 이는 일부 예에서 효율적이고 비용 효율적인 데이터 처리 해결 방안으로 이어질 수 있다. 스마트 저장 장치(예로서, 임베디드 프로세서를 포함하는 SSD)는 호스트 CPU 프로세서에 의해 수행 될 수 있는 데이터 처리 기능의 적어도 일부를 수행하기 위한 플랫폼을 제공 할 수 있다. 스마트 SSD 또는 유사한 저장 장치 내에서 이러한 데이터 처리 기능을 수행하는 것은 스토리지 네트워크에 대한 에너지 소비, 네트워크 대역폭, CPU주기, 메모리 등에 관한 이점을 제공할 수 있다.
- [0022] 일부 예들에서, 오프로드된 기능들은 저장 네트워크의 성능을 향상시키고, 전력 소비를 감소시키기 위해, 임베디드 프로세서(예로서, FPGA 또는 SSD 컨트롤러 또는 이산 보조 프로세서)를 포함하는 저장 장치에서 RTL(Register-Transfer Level) 로직, HLS(High-Level Synthesis) 또는 FPGA(Field-Programmable Gate Array) 로직을 사용하여 구현 될 수 있다. 그러나, RTL 또는 HLS를 사용하는 고성능 오프로드된 기능의 긴 개발 주기는 시스템 개발의 다른 활동을 차단하거나 상당히 느리게 할 수 있다. 본 개시의 일부 실시 예는 소프트웨어(펌웨어라고도 함) 기능으로써 임베디드 프로세서(예로서, SSD 내의) 에서 오프로드된 기능의 실행을 가능하게 할 수 있다. 소프트웨어로서 오프로드된 기능을 실행하면, 빠른 개발 시간이 달성될 수 있다. 소프트웨어 기반 오프로드된 기능은 호스트(예로서, 호스트 CPU)에서 이미 해당 기능을 수행하는 소프트웨어 코드를 재사용 할 수 있다.
- [0023] 저장 장치 또는 SSD의 임베디드 프로세서에서 소프트웨어 또는 펌웨어로서 오프로드된 기능을 실행하는 것은 호스트 소프트웨어 스택과 오프로드된 커널 들간의 인터페이스를 소프트웨어 커널들 및 다른 형태의 커널 구현과 동일하게 유지할 수 있다. 따라서, 초기의 빠른 오프로드가 소프트웨어 커널 형태에서 수행될 수 있고, 나중에 오프로드된 기능은 전체 호스트 소프트웨어 스택 및/또는 시스템 아키텍처를 크게 변경하거나 중단하지 않고 하드웨어에서 구현될 수 있다. 빠른 오프로드 및 실행을 통해, 더 높은 성능의 커널들이 개발되는 동안 다른 시스템 활동(예로서, 시스템 통합, 검증, 특성화 등)이 진행될 수 있다.
- [0024] 위에서 언급 한 바와 같이, 본 발명의 일부 실시 예는 효율적이고 비용 효율적인 데이터 처리를 위해, 애플리케이션 기능들을 FPGA 또는 SSD 컨트롤러 또는 이산 보조 프로세서와 같은 임베디드 프로세서를 포함하는 SSD와 같은 컴퓨터의(연산) 저장 장치로 오프로드하기 위한 방법 및 시스템을 포함한다. 일부 실시 예는 오프로드된 기능 대신에 호스트(예로서, 호스트 CPU)에 대해 프록시로서 기능 할 수 있는 FPGA 하드웨어 기반 브릿지 커널을 포함 할 수 있다. 브릿지 커널은 호스트 드라이버로부터의 인자(argument)를 받아 들여 그 인자들(예로서, 매개 변수 값들, 메모리 포인트들 등)을 임베디드 프로세서(예로서, FPGA 또는 SSD 컨트롤러 또는 이산 보조 프로세서)에서 소프트웨어 커널들로서 실행되는 오프로드된 기능으로 전송할 수 있다. 다른 실시 예에서의 하드웨어 및 펌웨어 조각들은 호스트상에서 실행되는 애플리케이션들과 소프트웨어 커널들 사이의 브릿지으로써 쓰일 수 있다. 그런 다음 이러한 소프트웨어 커널을 고성능을 위해 하드웨어로 이동하고, 나중에 호스트로 투명하게 이동할 수 있다. 그러므로, 브릿지 커널 호스트 인터페이스는 추가 수정없이 사용될 수 있으며, 이를 통해 컴퓨터의 저장 장치들(예로서, 스마트 SSD들)의 사용자는, 예로서 기존 애플리케이션 기능 코드 베이스를 저장 장치 내의 임베디드 프로세서(예로서, FPGA 또는 SSD 컨트롤러 또는 이산 보조 프로세서)로 포팅(porting)하여 사용 사례를 신속하게 개발할 수 있다.
- [0025] 스마트 SSD를 위한 이러한 용이한 사용 사례 개발은 여기에서 개시된 기술의 더 빠른 사용자 채택을 허용 할 수 있다. HLS 및 RTL을 사용하는 고성능 커널은 비교적 많은 개발 시간과 리소스를 사용할 수 있다. 본 개시의 예시적인 실시 예는 고성능 커널이 백그라운드에서 개발되는 동안 사용자가 다양한 태스크(예로서, 시스템 통합, 고객 데모, 시스템 검증 및 기타 태스크)를 수행 할 수 있게 한다. 또한, 브릿지 커널은 다른 커널로 비용 기반 라우팅을 수행 할 수 있다. 즉, 브릿지 커널은 호스트로부터 수신된 인자의 일부 기능과 임베디드 프로세서를 포함하는 저장 장치에 구성된 다양한 저장 또는 네트워킹 매개 변수들을 기반으로 다른 커널을 호출 할 수 있다는 것을 말한다. 예로서, 일부 실시 예들에서, 런타임 동안 호스트로부터 수신 된 인자에 기초하여, 브릿지 커널은 적절하게 상이한 성능 레벨 커널을 호출 할 수 있다.
- [0026] 도 1a는 브릿지 커널을 포함하는 시스템의 예시적인 블록도를 도시한다. 시스템(100)은 호스트(102) 및 저장 장치(104)(예로서, FPGA 프로세서를 포함하는 NVMe-oF(Non-Volatile Memory express over Fabrics) 호환 eSSD(Ethernet SSD), 또는 가속 기능이 있는 NVMe SSD와 같은 스마트 SSD)를 포함한다. 호스트(102)는 사용자 애플리케이션(들)(106) 모듈 및 OpenCL(open computing language, 108) 모듈을 포함한다. OpenCL(108) 모듈은 컴퓨팅 오프로드 애플리케이션 프로그래밍 인터페이스를 나타내기 위해 사용될 수 있다(예로서, 이는 호스트

(102)와 저장 장치(104) 사이의 임의의 컴퓨팅 오프로드 인터페이스(예로서, SSD)에 적용될 수 있다. 호스트(102)는 PCIe(Peripheral Component Interconnect Express) 또는 이더넷 연결을 사용하여 저장 장치(104)와의 연결을 설정할 수 있다.

[0027] 저장 장치(104)는 브릿지 커널을 포함한다. 일부 실시 예에서, 브릿지 커널은 브릿지 커널 하드웨어(110) 및 브릿지 커널 펌웨어(116)를 포함 할 수 있다. 저장 장치(104)는 임베디드 프로세서(112), FIFO(first in, first out, 118) 레지스터 및 DRAM(dynamic random-access memory)를 더 포함한다. DRAM(120)은 데이터 버퍼(122) 및 핸드셰이크 영역(124)을 포함한다. 임베디드 프로세서(112)는 오프로드된 애플리케이션 기능(들)(114) 및 브릿지 커널 펌웨어(116)를 포함한다.

[0028] 일부 실시 예에서, 시스템(100)의 호스트(102)는 운영 체제(OS) 및 파일 시스템(FS) 및 NVMe-oF 드라이버를 더 포함 할 수 있다. 일부 실시 예에서, 저장 장치(104)는 이더넷 스위치, 베이스보드 관리 컨트롤러(baseboard management controller, 이하 BMC) 및 PCIe 스위치를 포함하는 새시(도시되지 않음) 내에 있을 수 있다. 이더넷 스위치는 미드 플레인을 통해 저장 장치(104)(예로서, 스마트 SSD)에 대한 이더넷 연결을 제공하고, PCIe 스위치는 미드 플레인을 통해 저장 장치(104)에 관리 인터페이스를 제공한다. BMC는 시스템 관리자에 의해 주어진 명령(어)에 따라 저장 장치(104)를 프로그래밍 할 수 있다. BMC는 이더넷 스위치, PCIe 스위치 및 저장 장치(104)를 포함하는 새시의 내부 구성 요소를 관리 할 수 있다. BMC는 시스템 관리를 위한 PCIe 및/또는 시스템 관리 버스(SMBus) 인터페이스를 지원할 수 있다. BMC는 저장 장치(104)를 구성하고 이더넷 스위치를 프로그래밍 할 수 있다. 일부 실시 예들에서, 이더넷 스위치는 호스트(102)와 저장 장치(104) 사이의 네트워크 연결을 제공한다.

[0029] 도 1b는 브릿지 커널(예로서, 브릿지 커널 하드웨어(110))에 의한 오프로드된 기능들의 상이한 커널들로의 비용 기반 라우팅 프로세스를 도시한다. 일부 실시 예들에서, 시스템(100)은 오프로드된 기능들(114)(예로서, 오프로드된 애플리케이션 기능들)을 스마트 SSD(예로서, 저장 장치(104))와 같은 저장 장치로 비교적 빠르게 포팅 할 수 있다. 일부 실시 예들에서, 호스트(예로서, 호스트(102)) CPU상에서 실행되는 사용자 애플리케이션(들)(106)은 일부 기능들을 스마트 SSD(예로서, 저장 장치(104))로 오프로드하는 것으로부터 이익을 얻을 수 있다. 일반적으로, SSD(예로서, 저장 장치(104))에 저장된 많은 데이터를 처리하는 기능은 이러한 오프로드로부터 이익을 얻을 수 있다. 오프로드된 기능(예로서, 오프로드된 애플리케이션 기능(들)(114))은 "커널들"(예로서, 126(1), ..., 126(n), 126(n+1), ..., 126(m))로 언급될 수 있다.

[0030] 도 1b에서, 커널들(126(1), ..., 126(n))은 소프트웨어 커널들이고, 커널들(126(n+1), ..., 126(m))은 하드 커널들 또는 하드웨어 커널들이다. 커널들(126(1), ..., 126(n)) 가운데서 소프트웨어 커널은 임베디드 프로세서(112) 상에서 실행될 수 있다. 호스트 애플리케이션(예로서, 사용자 애플리케이션(들)(106))에서 실행(은 호스트(102) 측면 상의 OpenCL(예로서, OpenCL(108))과 같은 언어에 의해 제공되는 인터페이스를 통하고, 저장 장치(104) 측면 상의 브릿지 커널 하드웨어(110)를 통하여 커널들(126(1), ..., 126(n), 126(n+1), ..., 126(m))과 접속할 수 있다.

[0031] 일부 실시 예에서, 커널들(126(1), ..., 126(n), 126(n+1), ..., 126(m)) 중 복수의 커널들은 고성능 커널들(예로서, 고성능 커널은 설정된 또는 미리 결정된 임계 값을 초과하는 성능 메트릭(지표)을 가질 수 있음)이 될 수 있고, 커널들(126(1), ..., 126(n), 126(n+1), ..., 126(m)) 중 다른 복수의 커널들은 저성능 커널일 수 있다(예로서, 저 성능 커널은 설정된 또는 미리 결정된 임계 값 미만의 성능 메트릭을 가질 수 있음). 브릿지 커널(예로서, 브릿지 커널 하드웨어(110))은 데이터 처리를 위해 커널들(126(1), ..., 126(n), 126(n+1), ..., 126(m)) 중에서 하나 이상의 커널의 선택을 수행 할 수 있다.

[0032] 일부 실시 예들에서, 브릿지 커널은 비용 기능에 기초하여 데이터 처리를 위한 커널(들)의 선택을 수행 할 수 있다. 일부 실시 예에서, 비용 기능은 호스트(102)로부터 수신된 인자 및 호스트(102)로부터 수신된 스토리지 또는 네트워킹 매개 변수들에 기초하여 결정될 수 있다. 스토리지 또는 네트워킹 매개 변수의 예는 SQID(Submission Queue Identifier), CQID(Completion Queue Identifier), 스트림 ID, 호스트 ID, LBA(Logical Block Address) 범위, NSID(Network Service ID), MAC(Media Access Control) ID, TCP(Transmission Control Protocol)/IP(Internet Protocol) 필드, 애플리케이션 유형, 애플리케이션 식별자, 호스트(102)와 관련된 시간 및/또는 날짜, 이들의 조합 등을 포함할 수 있지만, 이에 한정되지는 않는다.

[0033] 예로서, 인자들은 호스트(102) 애플리케이션 측으로부터 브릿지 커널로 전달 될 수 있다(예로서, 호스트(102) 애플리케이션은 비용 기능을 브릿지 커널 하드웨어(110)로 오프로드 할 수 있다). 브릿지 커널은 주어진 커널을 사용하는 비용을 계산할 수 있다. 브릿지 커널은 비용 기능 및 호스트(102)로부터 수신 된 저장 또는 네트워킹

매개 변수들의 조합에 기초하여 최적의 커널을 선택할 수 있다. 일부 실시 예들에서, 예로서, 저장 장치(104)는 2개의 데이터 압축 커널을 구현할 수 있는데, 하나는 더 나은 처리량을 위해, 다른 하나는 더 나은 압축률을 위해 설계된 것이다. 호스트(102)로부터의 압축의 주어진 런타임 호출에 대해, 브릿지 커널은 길이 인자를 검사할 수 있고, 길이가 예를 들어 64KB보다 클 때마다 더 나은 압축률 커널을 선택할 수 있고 나머지를 위해 다른 유형의 압축 커널을 선택할 수 있다. 일부 실시 예에서, 브릿지 커널 하드웨어(110)는 FIFO 레지스터를 이용하여 데이터 처리에 사용하기 위해, 커널들(126(1), ..., 126(n), 126(n+1), ..., 126(m)) 중에서 최적의 커널(들)을 선택할 수 있다.

[0034] 도 2a 내지 2b는 오프로드된 커널과 접속하는 호스트 애플리케이션(예로서, 사용자 애플리케이션(들)(106))의 예시적인 흐름을 도시한다. 201에서, 브릿지 커널(예로서, 브릿지 커널 드라이버)은 상위 계층(예로서, 도 4에 도시된 바와 같이, 호스트 상위 계층 소프트웨어(136))으로부터 호출(call) 또는 호출(invocation)을 수신할 수 있다. 202에서, 브릿지 커널(예로서, 브릿지 커널 드라이버)은 호스트 인자들(예로서, 인자-1, ..., 인자-n)을 호스트 레지스터 인터페이스(예로서, 도 3)에 저장할 수 있다. 203에서, 브릿지 커널(예로서, 브릿지 커널 드라이버)은 데이터 처리를 위해 버퍼(예로서, 데이터 버퍼(122))를 할당할 수 있다. 204에서, 브릿지 커널(예로서, 브릿지 커널 드라이버)은 오프로드된 애플리케이션 기능(114)에 의해 필요한 다른 인자를 초기화할 수 있다. 205에서, 브릿지 커널(예로서, 브릿지 커널 드라이버)은 처리를 위해 버퍼로 데이터를 폐지할 수 있다. 206에서, 브릿지 커널(예로서, 브릿지 커널 드라이버)은 프로세싱 커널을 호출할 수 있다. 데이터 처리가 완료되면, 207에서, 시스템(100)은 클린 업 동작을 수행할 수 있다. 208에서, 오프로드된 애플리케이션 기능(들)(114)은 브릿지 커널(예로서, 브릿지 커널 드라이버)을 통해, 상위 계층 애플리케이션 소프트웨어에 대한 호출을 반환할 수 있다.

[0035] 일부 실시 예에서, 브릿지 커널은 호스트(102)에 일반 레지스터 인터페이스를 제공할 수 있다. 도 3은 브릿지 커널에 의해 호스트(102)에 제시된 레지스터 인터페이스(300)의 예를 도시한다. 레지스터 인터페이스(300)는 저장 장치(104)에 위치될 수 있다. 시스템(100)은 오프로드된 기능(예로서, 오프로드된 애플리케이션 기능(들)(114))에 인자를 전달하고 반환된 값 또는 상태를 검색하기 위해 (예로서, 도 3에 도시된 바와 같이) 레지스터 인터페이스(300)를 사용하여 커널들을 호스트 애플리케이션(예로서, 사용자 애플리케이션(들)(106)에서 실행중인)에 제시할 수 있다. 일부 실시 예에서, 시스템(100)은 커널에 대한 버퍼 할당을 용이하게 하기 위해 OpenCL(108) 플랫폼을 사용할 수 있다.

[0036] 도 4는 도 1의 DRAM(120)의 핸드셰이크 영역(124)의 레이아웃의 예를 도시한다. 도 4에 도시된 바와 같이, 일부 실시 예에서, 브릿지 커널 하드웨어(110)는 호스트(102)로부터 수신된 인자들(예로서, 인자-1, ..., 인자-n)을 저장 장치(104)(예로서, 스마트 SSD)의 DRAM(120) 내에 있는 설정 또는 미리 결정된 메모리 위치에 일시적으로 저장할 수 있다. 일부 실시 예들에서, 온칩 FIFO(118) 구조는 DRAM(120) 대신에 오프로드된 애플리케이션 기능(들)(114), 펌웨어 및 브릿지 커널 펌웨어(116)를 통합하는 임베디드 프로세서(112)에 인자들을 전달하는데 사용될 수 있다.

[0037] 레지스터 인터페이스(300)의 내용, 예를 들어 인자-1 내지 인자-n은 DRAM(120)의 핸드셰이크 영역(124)에 일시적으로 저장될 수 있다. 일단 인자가 DRAM(120)에 일단 일시적으로 저장되면, 브릿지 커널 하드웨어(110)는 DRAM(120)의 핸드셰이크 영역(124)에 준비(128) 플래그를 설정할 수 있다. 이 시점에서, 브릿지 커널 하드웨어(110)는 펌웨어(예로서, 브릿지 커널 펌웨어(116))에서 실행되는 오프로드된 기능이 처리를 완료하기를 기다릴 수 있다. 따라서, 오프로드된 기능 처리의 완료를 결정하기 위해, 브릿지 커널 하드웨어(110)는 DRAM(120)의 핸드셰이크 영역(124)에서 "완료"(130) 플래그를 폴링(polling)할 수 있다.

[0038] 브릿지 커널 하드웨어(110)가 "완료"(130) 플래그를 검출하면, 브릿지 커널 하드웨어(110)는 DRAM(120)의 핸드셰이크 영역(124)으로부터 임의의 에러 또는 다른 상태를 판독할 수 있다. 브릿지 커널 하드웨어(110)는 (도 3에 도시된 바와 같이) 레지스터 인터페이스(300)의 대응하는 레지스터로의 에러 및/또는 다른 상태(132)를 업데이트할 수 있다. 그러면 브릿지 커널 하드웨어(110)는 "완료"(130) 레지스터를 설정할 수 있으며, 이는 결국 오프로드된 기능 호출의 완료를 표시하기 위해 호스트 애플리케이션 소프트웨어(134)에 의해 폴링될 수 있다.

[0039] 도 5는 브릿지 커널 하드웨어를 상태 기계로서 사용하는 방법(500)을 도시한다. 브릿지 커널 하드웨어는 도 1의 브릿지 커널 하드웨어(110)일 수 있다.

[0040] 501에서, 브릿지 커널 하드웨어(110)는 유휴 상태로 유지된다. 502에서, 브릿지 커널 하드웨어(110)는 호스트(102)로부터 트리거를 수신한다. 503에서, 브릿지 커널 하드웨어(110)는 호스트 레지스터 인터페이스(예로서, 도 3)로부터, 호스트(102)로부터 수신된 인자(예로서, 인자-1, ..., 인자-n)를 폐지한다. 505에서, 브릿지 커널

하드웨어(110)는 DRAM(120)의 핸드셰이크 영역(124)에 호스트(102)로부터 수신 된 인자(예로서, 인자-1, ..., 인자-n)를 일시적으로 저장한다. 인자들이 DRAM(120)에 일시적으로 저장되면, 507에서, 브릿지 커널 하드웨어(110)는 DRAM(120)의 핸드셰이크 영역(124)에서 "준비"(128) 플래그를 설정한다.

[0041] 이 시점에서, 브릿지 커널 하드웨어(110)는 펌웨어(예로서, 브릿지 커널 펌웨어(116))에서의 오프로드된 기능이 인자에 기초하여 그것의 처리를 완료하기를 기다린다. 이를 위해, 509에서, 브릿지 커널 하드웨어(110)는 DRAM(120)의 핸드셰이크 영역(124)에서 "완료"(130) 플래그를 폴링한다. 일부 실시 예에서, 브릿지 커널 하드웨어(110)가 "완료"(130) 플래그를 검출하면, 브릿지 커널 하드웨어(110)는 DRAM(120)의 핸드셰이크 영역(124)로부터 임의의 에러 및/또는 다른 상태를 판독할 수 있다. 브릿지 커널 하드웨어(110)는 (도 3에 도시된 바와 같이) 레지스터 인터페이스(300)의 적절한 레지스터로 에러 및/또는 다른 상태(132)를 업데이트 한다. 브릿지 커널 하드웨어(110)는 오프로드된 기능 호출의 완료를 나타내기 위해 호스트 소프트웨어(134)에 의해 결국 폴링되는 "완료"(130) 레지스터를 설정한다.

[0042] 511에서, 브릿지 커널 하드웨어(110)는 호스트 레지스터 인터페이스(예로서, 도 3의 300)에서 준비 플래그를 설정한다. 511 이후에, 브릿지 커널 하드웨어(110)는 501로 복귀한다.

[0043] 도 6은 브릿지 커널 펌웨어 동작을 도시한다. 브릿지 커널 펌웨어는 도 1의 브릿지 커널 펌웨어(116) 일 수 있다.

[0044] 601에서, 브릿지 커널 펌웨어(116)는 유휴 상태로 유지된다. 603에서, DRAM(120)의 "준비" 플래그(128)는 임베디드 프로세서(112) 상에서 실행되는 브릿지 커널 펌웨어(116)에 의해 폴링된다. 다시 말해서, 브릿지 커널 펌웨어(116)는 적절한 인자를 갖는 오프로드된 기능(예로서, 오프로드된 애플리케이션 기능(114))를 부르는 오프로드 관리자로서 동작한다. 브릿지 커널 펌웨어(116)가 "준비" 플래그가 설정되는 것을 검출하면, 605에서, 브릿지 커널 펌웨어(116)는 DRAM(120)의 핸드셰이크 영역(124)에 일시적으로 저장된 인자들(예로서, 인자-1, ..., 인자-n)을 페치한다. 607에서, 브릿지 커널 펌웨어(116)는 호스트(102) 인자들과 함께 오프로드된 애플리케이션 기능(들)(114)를 호출한다. 오프로드된 애플리케이션 기능(들)(114)는 인자들(예로서, 인자-1, ..., 인자-n)을 사용하고 설계된대로 데이터 처리 기능들을 수행한다. 오프로드된 애플리케이션 기능(들)(114)의 예는 608에 도시된다. 데이터 처리가 완료되면, 609에서, 오프로드된 애플리케이션 기능(들)(114)은 브릿지 커널 펌웨어(116)에 대한 호출을 반환한다. 611에서 브릿지 커널 펌웨어(116)는 DRAM(120)의 핸드셰이크 영역(124)에서 "완료" 플래그를 설정한다. "완료" 플래그는 앞서 설명된 바와 같이 브릿지 커널 하드웨어(110)에 의해 폴링되어 완료를 다시 호스트(102)로 전송한다. 611 이후에, 브릿지 커널 펌웨어(116)는 601로 되돌아 간다.

[0045] 일부 실시 예에서, 브릿지 커널은 호스트 애플리케이션(예로서, 사용자 애플리케이션(들)(106)에서 실행)에 대한 프록시로서 동작한다. 예로서, 호스트 애플리케이션(예로서, 사용자 애플리케이션(들)(106)에서 실행중인)은 오프로드된 기능이 어디서 또는 어떻게 구현되는지와 무관하다(알지 못한다). 일부 실시 예에서, 커널 구현은 호스트(102)에 투명 할 수 있다. 일부 실시 예에서, 커널은 소프트웨어 커널로서 임베디드 프로세서(예로서, 임베디드 프로세서(112)) 상에 구현 될 수 있다. 그러나, 일부 실시 예에서, 커널은 하드웨어 게이트들(예로서, 브릿지 커널 하드웨어(110))로 구현 될 수 있다.

[0046] 일부 실시 예들에서, 브릿지 커널 자체는 오프로드된 기능(들)(예로서, 오프로드된 애플리케이션 기능(들)(114))에 대해 무관할 수 있으며, 이는 사용자 기밀성(예로서, IP 주소 기밀성)을 가능하게 할 수 있다. 따라서, 사용자는 저장 장치 사용자에게 공개하지 않고도 소프트웨어 커널 또는 하드웨어 커널로 자체 오프로드된 기능을 만들 수 있다. 즉, 브릿지 커널 메커니즘의 예시적인 실시 예는 개인(비공개, private) 가속 기능을 저장 장치에 오프로드하기 위해 사용자에게 의해 사용될 수 있는 플랫폼으로서 동작할 수 있다. 일부 실시 예에서, 커널 및/또는 오프로드된 기능에 대한 호스트 인터페이스는 변경되지 않을 수 있다(예로서, 상당한 호스트 측 변경없이 커널을 고성능 버전으로 업그레이드 할 수 있다). 그러한 경우에, 브릿지 커널(예로서, 브릿지 커널 하드웨어(110)) RTL은 호스트 애플리케이션(예로서, 사용자 애플리케이션(들)(106))에 표준 또는 합의된 인터페이스를 제공 할 수 있다.

[0047] 일부 실시 예에서, 브릿지 커널(예로서, 브릿지 커널 하드웨어(110))은 브릿지 커널 펌웨어(116)와의 핸드셰이크를 수행하기 위해 FPGA DRAM(예로서, DRAM 120) 메모리를 사용할 수 있다. 다른 실시 예에서, 브릿지 커널 하드웨어(110)는 브릿지 커널 펌웨어(116)와의 핸드셰이크를 수행하기 위해 온칩 FIFO(예로서, FIFO(118))를 사용할 수 있다. 일부 실시 예에서, 브릿지 커널(예로서, 브릿지 커널 하드웨어(110)) RTL은 임베디드 프로세서(112)상에서 실행되는 오프로드된 기능에 인자를 전달하기 위해 FPGA DRAM(예로서, DRAM(120))을 사용할 수 있다.

- [0048] 일부 실시 예에서, 브릿지 커널(예로서, 브릿지 커널 하드웨어(110))은 기본(기저의) 전송 메커니즘(예로서, PCIe, 이더넷 등)과 관련하여 무관할 수 있으며, 이(기본(기저의) 전송 메커니즘)는 호스트(102)와 저장 장치(104) 간의 연결을 설정하기 위해 사용된다. 일부 실시 예에서, 브릿지 커널(예로서, 브릿지 커널 하드웨어(110))을 포함하는 시스템(100)은 브릿지 커널을 사용하지 않는 시스템과 비교하여 비교적 빠른 포팅 또는 개발 옵션을 제공함으로써 비교적 빠른 사용자 참여를 제공 할 수 있고, 사용자들에 의해 기존 코드 베이스의 재사용을 증가시킬 수 있고, 고성능 커널을 개발되는 동안 시스템이 다른 활동을 수행하게 할 수 있다. 일부 실시 예들에서, 브릿지 커널(예로서, 브릿지 커널 하드웨어(110))을 포함하는 시스템(100)은 또한 시스템 통합, 인증, 테스트, 데모 등을 제공 할 수 있다.
- [0049] 일부 실시 예들에서, 브릿지 커널은 호스트에 의해 전달된(및/또는 장치에 의해 설정되는) 인자들을 사용하여 호출 할 오프로드된 기능(예로서, 커널)를 결정할 수 있다. 일부 실시 예에서, 브릿지 커널은 소프트웨어 커널, 하드 커널 또는 다양한 성능 레벨의 하나 이상의 커널을 호출 할 수 있다. 일부 실시 예에서, 브릿지 커널은 인자에 기초하여 특정 커널을 선택하기 위해 비용 기능을 사용할 수 있다. 일부 실시 예들에서, 커널 선택은 애플리케이션 유형, 애플리케이션 식별자, 네임스페이스 식별자, 호스트 식별자, LBA 어드레스 범위, NVMe 세트 식별자, NVMe 제출 큐 식별자, 완료 큐 식별자, 스트림 식별자, 이더넷 MAC 식별자, TCP/IP 주소 및 기타 전송/네트워크 매개 변수들과 같은 구성 매개 변수들 및 날짜 및 시간과 같은 일반 매개 변수에 기초한다.
- [0050] 비록 여기에서 제 1, 제 2, 제3 등의 용어들은 다양한 요소들, 성분들, 영역들, 층들 그리고/또는 섹션들을 설명하기 위해 사용되지만, 이러한 요소들, 성분들, 영역들, 층들 그리고/또는 섹션들은 이러한 용어들로 인해 제한되지 않는 것으로 이해될 것이다. 이러한 용어들은 다른 요소, 성분, 영역, 층, 또는 섹션으로부터 하나의 요소, 구성, 영역, 층 또는 섹션을 구별하기 위해 사용된다. 따라서, 후술하는 제 1 구성 요소, 성분, 영역, 층, 또는 섹션은 본 발명의 사상 및 범위를 벗어나지 않고, 제 2 구성 요소, 성분, 영역, 층, 또는 섹션을 지칭 할 수 있다.
- [0051] 하나의 요소 또는 도면에서 도시된 다른 구성 요소(들) 또는 특징(들)과의 특징적인 관계를 설명하기 위한 설명을 용이하게 하기 위해 “아래의”, “아래”, “낮은”, “특정 부분 아래”, “위에”, “상부”와 같은 공간적이고 상대적인 용어들이 여기에서 사용될 수 있다. 공간적이고 상대적인 용어들은 도면에서 묘사된 방향에 대해 사용 또는 동작에서 장치의 다른 방향들을 포함하도록 의도된 것이 이해될 것이다. 예를 들면, 만약 도면의 장치가 뒤집어지면, 다른 구성 요소들 또는 특징들의 “아래” 또는 “아래의” 또는 “특정 부분 아래”로 설명된 구성요소들은 다른 구성 요소들 또는 특징들의 “위로” 맞춰지게 된다. 따라서, “아래의” 또는 “특정 부분 아래”의 예시적인 용어들은 위 또는 아래 방향 모두를 포함할 수 있다. 장치는 다르게 맞춰질 수 있으며 (예를 들면, 90도 또는 다른 방향으로 회전됨) 그리고 공간적으로 상대적인 기술어들은 그에 따라 해석되어야 한다. 또한, 요소 또는 층이 두 개의 요소들 또는 층들 사이로 언급되는 때, 그것은 단지 요소 또는 층이 두 요소들 또는 층들 사이에 있을 수 있거나, 또는 하나 또는 그 이상의 사이의 요소들 또는 층들이 또한 존재할 수 있다.
- [0052] 본 명세서에서 사용된 용어들은 단지 특정한 실시 예들을 설명하기 위한 것이고, 본 발명을 제한하려는 것으로 의도되지 않았다. 본 명세서에서 사용된 바와 같이, “대체로”, “약” 용어 그리고 이와 유사한 용어들은 근사치의 용어들로서 사용되고, 정도의 용어들로서 사용되지 않고, 본 발명의 당업자에 의해 식별되는 측정된 또는 계산된 값들의 고유한 변동을 고려하기 위한 것이다.
- [0053] 본 명세서에서 사용된 바와 같이, 문맥상 명백하게 다르게 뜻하지 않는 한, 단수 형태 “하나”는 복수의 형태도 포함하는 것으로 의도된다. “구성되는”, “구성되고 있는”, “포함하는”, 그리고 “포함하고 있는” 용어들이 본 명세서에서 사용될 때, 이러한 용어들은 정해진 특징들, 정수들, 단계들, 동작들, 요소들, 그리고/또는 성분들이 존재를 명시하나, 하나 또는 그 이상의 다른 특징들, 정수들, 단계들, 동작들, 요소들, 성분들, 그리고/또는 그것들의 그룹들의 추가 또는 존재를 불가능하게 하지 않는다. 본 명세서에서 사용된 바와 같이, “그리고/또는” 용어는 하나 또는 그 이상의 열거된 항목들과 연관된 임의의 그리고 모든 조합을 포함한다. “적어도 하나”와 같은 표현들은 요소들 전체 리스트를 수정하고 그리고 리스트의 개별 요소들을 수정하지 않는다. 또한, 본 발명의 실시 예들을 기술할 때 “할 수 있다”의 사용은 “본 발명의 하나 이상의 실시 예들”을 의미한다. 본 명세서에서 사용된 바와 같이, “사용”, “사용되는”, 그리고 “사용된” 용어들은 “이용”, “이용되는”, 그리고 “이용된” 용어들의 동의어로 각각 간주 될 수 있다. 또한, “예시” 용어는 예 또는 그림을 의미한다.
- [0054] 하나의 구성 요소가 다른 구성과 '연결'된다고 언급될 때에는 그것이 다른 구성 요소와 직접적으로 연결되거나,

중간의 매개 요소가 존재할 수 있는 것으로 이해되어야 한다. 반면, 하나의 구성 요소가 다른 구성 요소와 “직접적으로 연결” 된다고 언급되는 경우에는, 중간에 매개 요소가 존재하지 않음을 나타낸다. 여기서 사용되는, “및/또는” 용어는 하나 또는 그 이상의 관련된 나열된 사항의 모든 가능한 조합을 지시하거나 포함하는 것으로 이해되어야 한다.

[0055] 여기에서 인용된 임의의 수치 범위는 인용 범위 내에 포함 된 동일한 수치 정밀도의 모든 하위 범위들을 포함하는 것으로 의도된다. 예를 들어, "1.0 내지 10.0"의 범위는 언급 된 최소값 1.0과 언급 된 최대 값 10.0 사이, 즉 2.4 내지 7.6과 같은 10.0 이하의 최대 값 및 1.0 이상의 최소값을 갖는(및 포함하는) 모든 하위 범위를 포함하도록 의도된다. 예를 들어, 본 명세서에 인용 된 임의의 최대 수치 제한은 그 안에 포함 된 모든 하위 수치 제한을 포함하는 것으로 의도되고, 본 명세서에 인용 된 임의의 최소 수치 제한은 그에 포함된 모든 더 높은 수치 제한을 포함하는 것으로 의도된다.

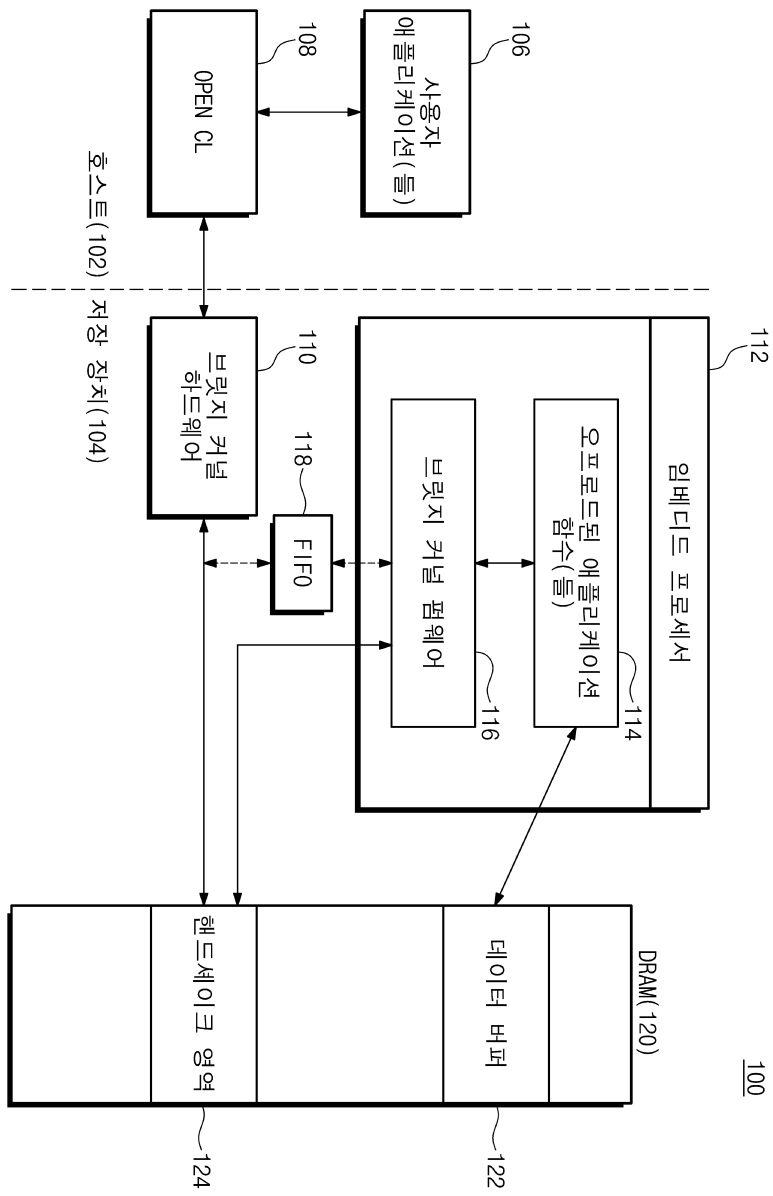
[0056] 일부 실시 예에서, 본 개 시의 방법 및 시스템의 상이한 실시 예의 하나 이상의 출력은 본 개시의 방법 및 시스템의 상이한 실시 예의 하나 이상의 출력 또는 하나 이상의 출력에 관한 정보를 디스플레이 하기 위한 디스플레이 장치에 연결되거나 이를 갖는 전자 장치로 전송될 수 있다.

[0057] 본 명세서에서 기술된 본 발명의 실시 예에 따른 전자 또는 전기 장치들 그리고/또는 다른 임의의 관련된 장치 들 또는 요소들은 임의의 적합한 하드웨어, 펌웨어(예를 들면, Application Specific Integrated Circuit; ASIC), 소프트웨어, 또는 소프트웨어, 펌웨어, 그리고 하드웨어의 조합을 이용하여 구현될 수 있다. 예를 들어, 이러한 장치들의 다양한 요소들은 하나의 집적 회로(Integrated Circuit; IC) 칩 또는 분리된 IC 칩들로 형성될 수 있다. 또한, 이러한 장치들의 다양한 요소들은 유연한 인쇄 회로 필름(Flexible Printed Circuit Film), TCP(Tape Carrier Package), 인쇄 회로 기판(Printed Circuit Board; PCB) 위에 구현되거나 하나의 기판 위에서 형성될 수 있다. 또한, 이러한 장치들의 다양한 요소들은 컴퓨터 프로그램 명령들을 실행하고 본 명세서에서 설명된 다양한 기능들을 수행하기 위한 다른 시스템 요소들과 상호 작용하는 하나 이상의 컴퓨팅 장치들에서 또는 하나 이상의 프로세서들에서 수행되는 프로세스 또는 스레드(Thread)일 수 있다. 컴퓨터 프로그램 명령들은 예를 들면 RAM(Random Access Memory)과 같은 표준 메모리 장치를 이용하는 컴퓨팅 장치에서 구현되는 메모리 내에 저장된다. 컴퓨터 프로그램 명령들은 또한 예를 들면 CD-ROM, 플래시 드라이브(Flash Drive), 또는 그와 같은 다른 일시적이지 않은 컴퓨터 읽기 가능한 미디어(Non-transitory Computer Readable Media)에 저장될 수도 있다. 또한, 본 발명의 당업자는 본 발명의 예시적인 실시 예들의 사상 및 범위를 벗어나지 않고 다양한 컴퓨팅 장치들의 기능은 단일 컴퓨팅 장치에 통합되거나 집적되고, 특정 컴퓨팅 장치의 기능이 하나 또는 그 이상의 다른 컴퓨팅 장치들에 분산될 수 있음을 인식해야 한다.

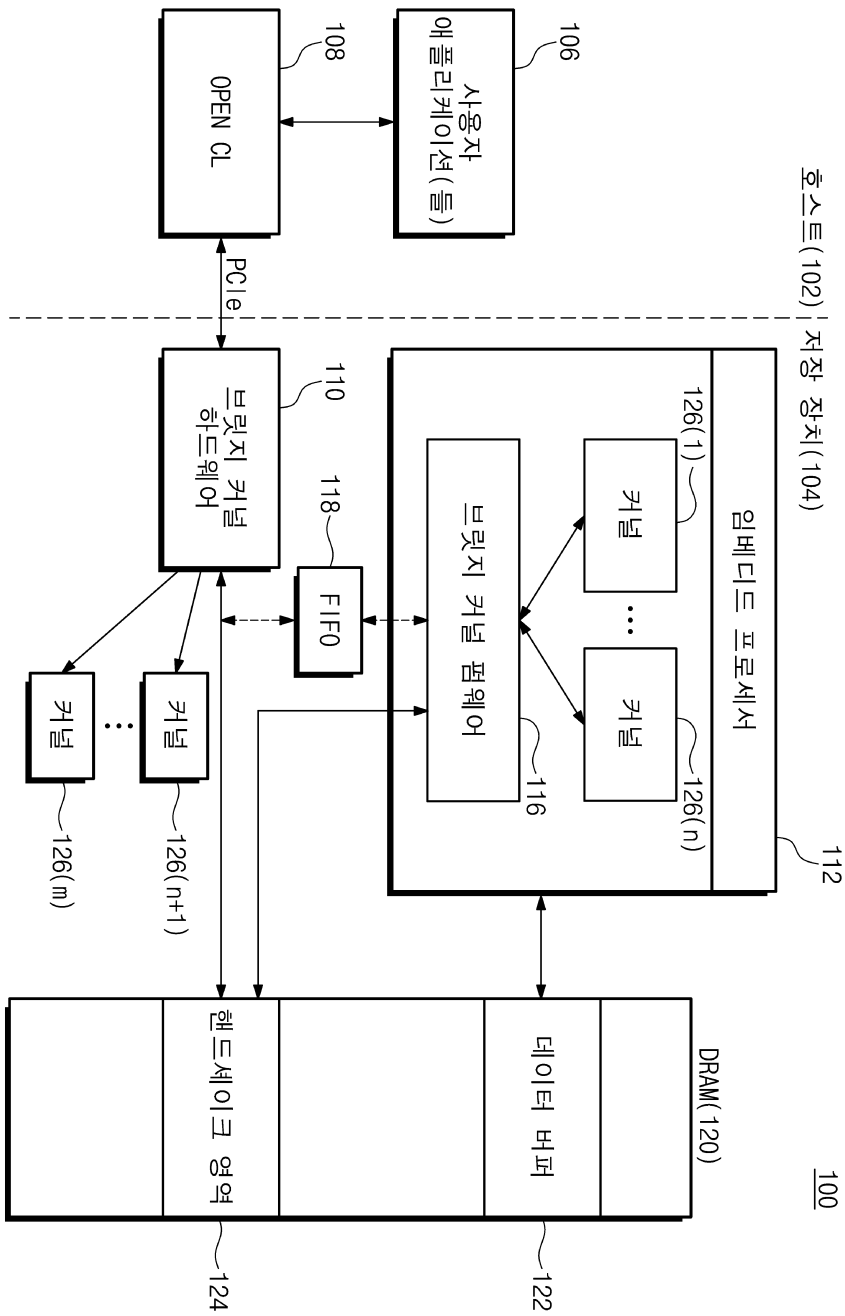
[0058] 애플리케이션 기능을 장치로 오프로드하기 위한 시스템 및 방법의 예시적인 실시 예가 여기에서 구체적으로 설명되고 도시되었지만, 많은 수정 및 변형이 당업자에게 명백할 것이다. 따라서, 여기에서 개시된 따라서, 본 개 시의 원리에 따라 구성된 장치에 애플리케이션 기능을 오프로드하는 시스템 및 방법은 본 명세서에 구체적으로 설명 된 것과 다르게 구현될 수 있음이 이해되어야 한다. 본 발명 개념은 또한 아래의 청구 범위 및 그 등가물에서 정의된다.

도면

도면1a



도면1b



도면2a

```

/***** OPEN CL BASED HOST FUNC_SMARTSSD *****/

RECEIVE CALL FROM UPPER LAYER
RECEIVE CALL FROM THE UPPER LAYER,
RETRIEVE ARGUMENTS

ALLOCATE BUFFERS FOR DATA PROCESSING
PTR1=ALLOC BUFFER (SIZE 1); //THERE CAN BE MANY DIFFERENT METHODS TO
ALLOCATE BUFFERS, BEYOND THE SCOPE
PTR2=ALLOC_BUFFER (SIZE 2);

PTRN=ALLOC_BUFFER (SIZE N);

INITIAL OTHER ARGUMENTS NEEDED BY THE OFFLOADED FUNCTION
VAL1=1;
VAL2=2;

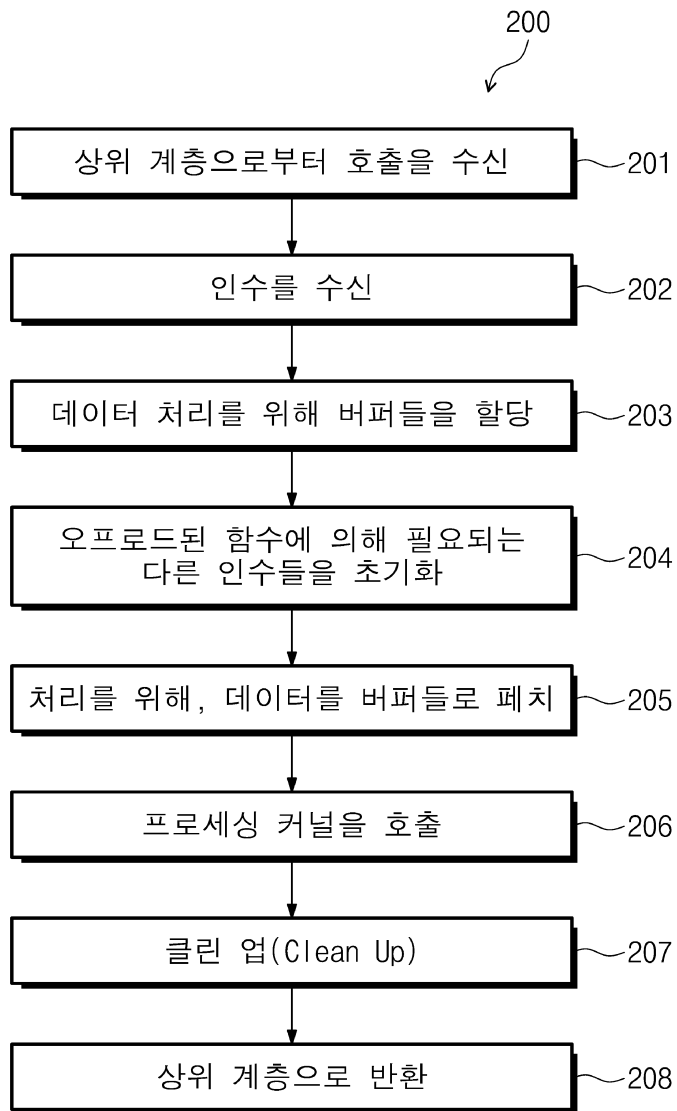
VALN=N;

FETCH DATA INTO THE BUFFERS FOR PROCESSING
FREAD(FPTR, PTR1); //THIS EXAMPLE USES FILE IO, THERE COULD BE VARIOUS
OTHER MEANS OF LOADING DATA INTO THE BUFFERS FOR PROCESSING

INVOLK THE PROCESSING KERNEL
STATUS=BRIDGE_KERNEL (VAL1, ...VALN, PTR1, ...PTRN);

CLEAN-UP
WHILE (BRIDGE_KERNEL NOT DONE)
RETURN TO UPPER LAYER
    
```

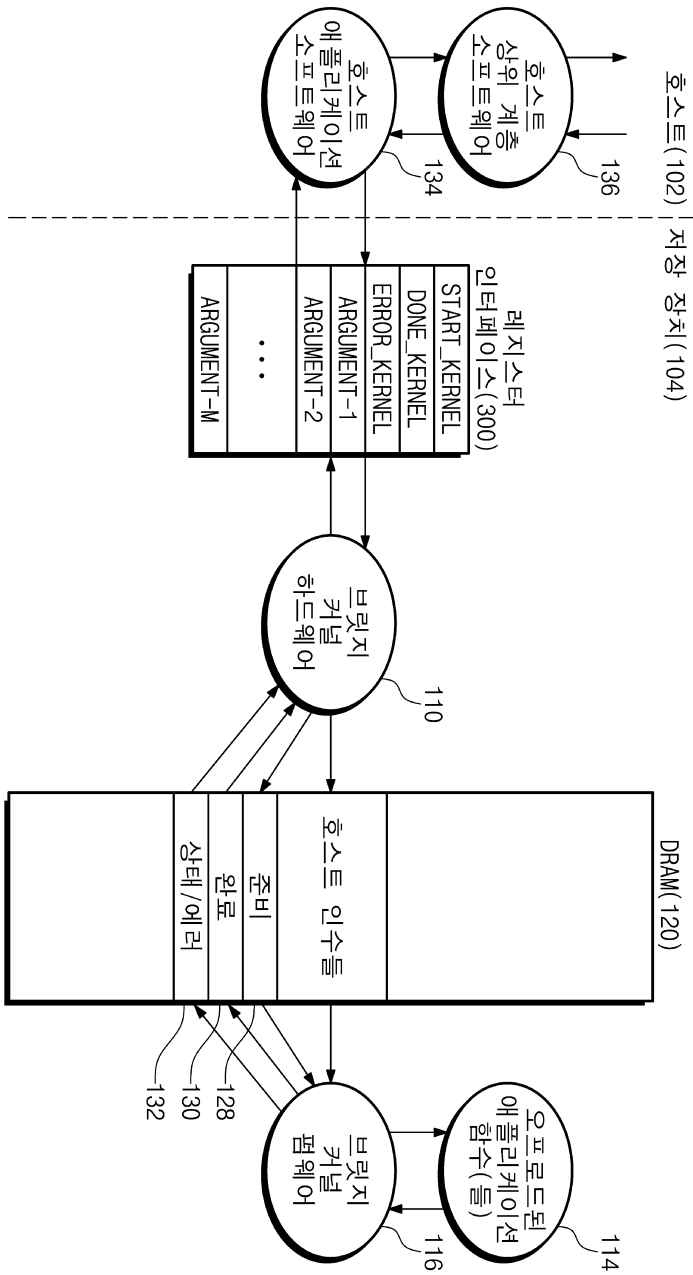
도면2b



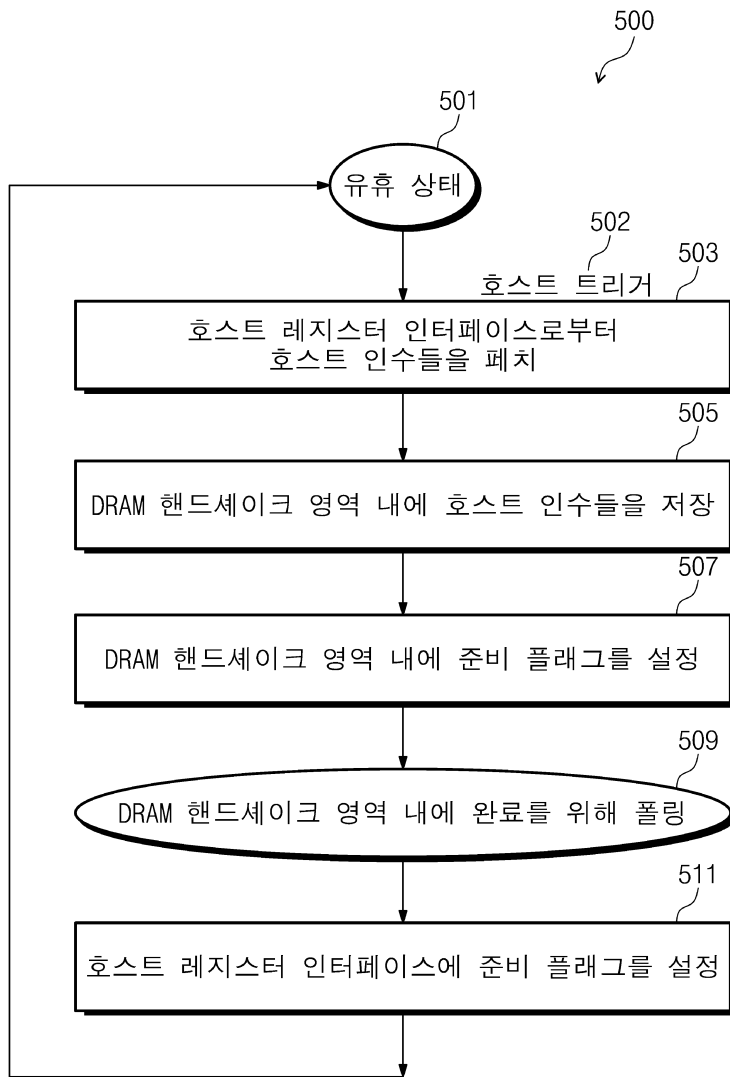
도면3

레지스터 #	레지스터 이름	주석
0	START_KERNEL	WRITE TO THIS REGISTER TRIGGER KERNEL
1	DONE_KERNEL	SET BY THE KERNEL AFTER PROCESSING IS COMPLETE, POLLED BY HOST
2	ERROR_KERNEL	SET BY THE KERNEL IF APPROPRIATE
3	ARGUMENT-1	ARGUMENT PASSE TO THE OFFLOADED FUNCTION, READ/WRITE BY HOST/SMART SSD
4	ARGUMENT-2	
⋮	⋮	
K	ARGUMENT-M	

도면4



도면5



도면6

