



등록특허 10-2759334



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2025년01월22일
(11) 등록번호 10-2759334
(24) 등록일자 2025년01월20일

- (51) 국제특허분류(Int. Cl.)
G06F 9/50 (2018.01) *G06N 20/00* (2019.01)
G06N 3/04 (2023.01) *G06N 3/063* (2023.01)
- (52) CPC특허분류
G06F 9/5044 (2013.01)
G06F 9/5016 (2013.01)
- (21) 출원번호 10-2022-7007076
- (22) 출원일자(국제) 2020년09월08일
심사청구일자 2022년03월02일
- (85) 번역문제출일자 2022년03월02일
- (65) 공개번호 10-2022-0038497
- (43) 공개일자 2022년03월28일
- (86) 국제출원번호 PCT/US2020/049648
- (87) 국제공개번호 WO 2021/101617
국제공개일자 2021년05월27일

(30) 우선권주장
62/938,304 2019년11월20일 미국(US)
16/720,717 2019년12월19일 미국(US)

(56) 선행기술조사문헌
미국공개특허 제2019-0312772호(2019.10.10.) 1
부.*

*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 18 항

심사관 : 김종기

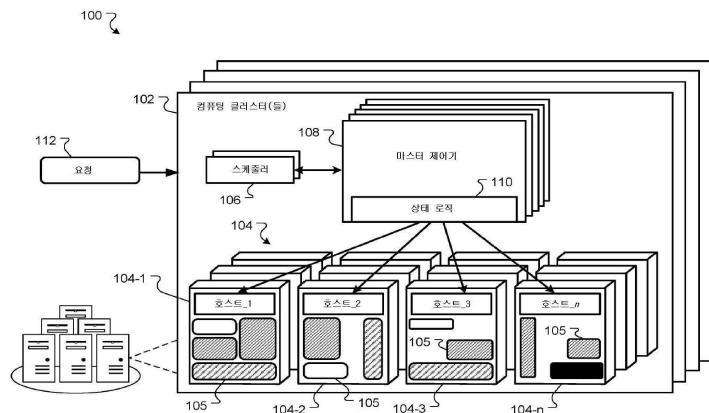
(54) 발명의 명칭 기계-학습 워크로드들에 대한 작업 스케줄링

(57) 요 약

ML 워크로드들의 작업들을 스케줄링하기 위한, 컴퓨터 저장 매체에 인코딩되는 컴퓨터 프로그램들을 포함해서 방법들, 시스템들 및 장치가 설명된다. 시스템은 워크로드들을 수행하라는 요청들을 수신하고, 요청에 기초하여, 워크로드들을 수행하기 위해 리소스 요구들을 결정한다. 시스템은 다수의 호스트들을 포함하고, 각각의 호스트

(뒷면에 계속)

대 표 도



는 다수의 가속기들을 포함한다. 시스템은 각각의 호스트에 대한 가속기들 및 리소스 요건들에 기초하여, 워크로드의 작업들을 실행하도록 배정되는 호스트들의 수량을 결정한다. 그 수량의 호스트들의 각각의 호스트에 대해, 시스템은 호스트의 메모리 액세스 토폴로지에 기초하여 작업 사양을 생성한다. 사양은 다수의 가속기들을 포함하는 호스트의 리소스들을 사용하여 호스트에서 실행될 작업을 명시한다. 시스템은 작업 사양들을 호스트들에 제공하며, 그리고 각각의 호스트가 호스트에 대한 작업 사양에 명시된 배정된 작업들을 실행할 때, 워크로드들을 수행한다.

(52) CPC특허분류

- G06N 20/00* (2021.08)
G06N 3/044 (2023.01)
G06N 3/045 (2023.01)
G06N 3/063 (2013.01)
G06F 2209/502 (2013.01)
G06F 2212/2542 (2013.01)
-

명세서

청구범위

청구항 1

복수의 신경망 레이어들을 포함하는 신경망을 구현하도록 각각 구성되는 하드웨어 가속기들을 사용하여, 기계-학습 워크로드를 수행하기 위해 작업들을 스케줄링하고 리소스들을 할당하기 위한 방법으로서, 상기 방법은,

상기 기계-학습(ML) 워크로드를 수행하라는 요청을 수신하는 단계;

상기 요청에 기초하여, 복수의 호스트들을 포함하는 분산 처리 시스템에서 상기 ML 워크로드를 수행하기 위해 리소스 요건을 결정하는 단계 – 상기 복수의 호스트들의 각각의 호스트는 각각의 복수의 하드웨어 가속기들을 포함함 –;

상기 각각의 호스트에 대한 각각의 복수의 하드웨어 가속기들 및 상기 리소스 요건에 기초하여, 상기 ML 워크로드를 형성하는 작업들의 세트로부터의 각각의 작업을 실행하도록 각각 배정되는 호스트들의 수량을 결정하는 단계;

상기 수량의 호스트들의 각각의 호스트에 대해:

상기 복수의 호스트들의 각각의 호스트에 대한 불균일 메모리 액세스(NUMA) 소켓들의 맵핑을 설명하는 시스템 파일에 기초하여, 상기 호스트의 리소스들을 결합하는 하나 이상의 소켓들을 식별하는 단계;

상기 호스트의 리소스들을 결합하는 상기 하나 이상의 소켓들에 기초하여, 상기 호스트의 제어 그룹을 형성하는 단계;

상기 호스트의 메모리 액세스 토폴로지에 기초하여, 상기 각각의 복수의 하드웨어 가속기들을 포함하는 상기 호스트의 리소스들을 사용하여, 상기 호스트에서 실행되도록 배정된 작업을 명시하는 각각의 작업 사양을 생성하는 단계; 및

상기 수량의 호스트들의 상기 호스트에 상기 각각의 작업 사양을 제공하는 단계; 및

상기 수량의 호스트들의 각각의 호스트에 의해, 상기 호스트에 대한 각각의 작업 사양에 명시된 작업을 실행함으로써 상기 ML 워크로드를 수행하는 단계를 포함하는, 방법.

청구항 2

제1 항에 있어서,

상기 각각의 호스트의 메모리 액세스 토폴로지는 상기 호스트에 대해 로컬적인 각각의 메모리를 포함하는 각각의 NUMA 토폴로지를 포함하고; 그리고

상기 각각의 메모리는 상기 각각의 메모리를, 상기 각각의 복수의 하드웨어 가속기들 중 각각의 하드웨어 가속기 및 상기 호스트의 하나 이상의 다른 리소스들에 결합시키는 소켓 인터페이스를 포함하는, 방법.

청구항 3

제2 항에 있어서,

상기 각각의 작업 사양에 명시된 작업을 실행하는 것은,

다수의 신경망 컴퓨터이션들의 각각의 부분들을 상기 각각의 복수의 하드웨어 가속기들의 각각의 하드웨어 가속기에 배정하는 것에 응답하여, 상기 복수의 신경망 레이어들 중 각각의 신경망 레이어에 대한 출력을 생성하도록 상기 다수의 신경망 컴퓨터이션들을 수행하는 것을 포함하는, 방법.

청구항 4

제2 항에 있어서,

상기 ML 워크로드를 수행하는 단계는,

상기 호스트의 제어 그룹의 각각의 리소스를 사용하고, 상기 각각의 메모리, 상기 하드웨어 가속기 및 상기 호스트의 리소스들 사이에 포함되는 각각의 프로세서 사이에서 교환되는 데이터에 기초하여, 상기 각각의 작업 사양에 대한 명령들을 처리하는 단계를 포함하는, 방법.

청구항 5

제4 항에 있어서,

상기 ML 워크로드를 수행하는 단계는,

상기 호스트의 제어 그룹의 각각의 리소스를 연결하는 하드웨어 소켓을 통해 교환되는 데이터에 기초하여 상기 명령들을 처리하는 것에 응답하여, 상기 각각의 작업 사양에 명시되는 작업들을 실행하는 단계를 포함하고,

상기 하드웨어 소켓은 상기 호스트에 의해 관리되는 다수의 리소스들 사이에서 공유되는 로컬 통신 버스를 규정하는, 방법.

청구항 6

제4 항에 있어서,

제1 호스트에 대한 각각의 NUMA 토폴로지는,

- i) 상기 제1 호스트에 로컬적인 리소스들의 각각의 구성의 각각의 제1 메모리; 및
- ii) 제2 다른 호스트에 로컬적이지만 상기 제1 호스트에 원격인 리소스들의 각각의 구성의 각각의 제2 다른 메모리에 부분적으로 기초하는, 방법.

청구항 7

제2 항에 있어서,

상기 호스트들의 수량을 결정하는 단계는,

상기 복수의 호스트들의 각각의 호스트에 의해 관리되는 리소스들의 구성을 설명하는 시스템 파일을 취득하는 단계; 및

상기 복수의 호스트들의 각각의 호스트에 대한 시스템 파일에 설명되는 상기 리소스들의 구성에 기초하여, 상기 호스트들의 수량을 결정하는 단계를 포함하는, 방법.

청구항 8

삭제

청구항 9

제1 항에 있어서,

상기 호스트의 제어 그룹의 가속기들에 대한 하나 이상의 소켓 인터페이스들에 기초하여, 상기 작업 사양의 ML 작업을 상기 제어 그룹에 배정하는 단계 – 상기 소켓 인터페이스들은 상기 시스템 파일에 설명되는 NUMA 소켓들의 맵핑에 포함됨 –; 및

상기 제어 그룹의 가속기들을 사용하여, 상기 제어 그룹 하의 공정으로서 상기 ML 작업을 실행하는 단계를 포함하는, 방법.

청구항 10

복수의 신경망 레이어들을 포함하는 신경망을 구현하도록 각각 구성되는 하드웨어 가속기들을 사용하여, 기계-학습 워크로드를 수행하기 위해 작업들을 스케줄링하고 리소스들을 할당하도록 구성되는 시스템으로서,

하나 이상의 처리 디바이스들; 및

동작들의 수행을 야기하도록 상기 하나 이상의 처리 디바이스들에 의해 실행 가능한 명령들을 저장하는 하나 이상의 비일시적 기계-판독 가능한 저장 디바이스들을 포함하며,

상기 동작들은,

상기 기계-학습(ML) 워크로드를 수행하라는 요청을 수신하는 동작;

상기 요청에 기초하여, 복수의 호스트들을 포함하는 분산 처리 시스템에서 상기 ML 워크로드를 수행하기 위해 리소스 요건을 결정하는 동작 – 상기 복수의 호스트들의 각각의 호스트는 각각의 복수의 하드웨어 가속기들을 포함함 –;

상기 각각의 호스트에 대한 각각의 복수의 하드웨어 가속기들 및 상기 리소스 요건에 기초하여, 상기 ML 워크로드를 형성하는 작업들의 세트로부터의 각각의 작업을 실행하도록 각각 배정되는 호스트들의 수량을 결정하는 동작;

상기 수량의 호스트들의 각각의 호스트에 대해:

상기 복수의 호스트들의 각각의 호스트에 대한 불균일 메모리 액세스(NUMA) 소켓들의 맵핑을 설명하는 시스템 파일에 기초하여, 상기 호스트의 리소스들을 결합하는 하나 이상의 소켓들을 식별하는 동작;

상기 호스트의 리소스들을 결합하는 상기 하나 이상의 소켓들에 기초하여, 상기 호스트의 제어 그룹을 형성하는 동작;

상기 호스트의 메모리 액세스 토폴로지에 기초하여, 상기 각각의 복수의 하드웨어 가속기들을 포함하는 상기 호스트의 리소스들을 사용하여, 상기 호스트에서 실행되도록 배정된 작업을 명시하는 각각의 작업 사양을 생성하는 동작; 및

상기 수량의 호스트들의 상기 호스트에 상기 각각의 작업 사양을 제공하는 동작; 및

상기 수량의 호스트들의 각각의 호스트에 의해, 상기 호스트에 대한 각각의 작업 사양에 명시된 작업을 실행함으로써 상기 ML 워크로드를 수행하는 동작을 포함하는, 시스템.

청구항 11

제10 항에 있어서,

상기 각각의 호스트의 메모리 액세스 토폴로지는 상기 호스트에 대해 로컬적인 각각의 메모리를 포함하는 각각의 NUMA 토폴로지를 포함하고; 그리고

상기 각각의 메모리는 상기 각각의 메모리를, 상기 각각의 복수의 하드웨어 가속기들 중 각각의 하드웨어 가속기 및 상기 호스트의 하나 이상의 다른 리소스들에 결합시키는 소켓 인터페이스를 포함하는, 시스템.

청구항 12

제11 항에 있어서,

상기 각각의 작업 사양에 명시된 작업을 실행하는 것은,

다수의 신경망 컴퓨테이션들의 각각의 부분들을 상기 각각의 복수의 하드웨어 가속기들의 각각의 하드웨어 가속기에 배정하는 것에 응답하여, 상기 복수의 신경망 레이어들 중 각각의 신경망 레이어에 대한 출력을 생성하도록 상기 다수의 신경망 컴퓨테이션들을 수행하는 것을 포함하는, 시스템.

청구항 13

제11 항에 있어서,

상기 ML 워크로드를 수행하는 동작은,

상기 호스트의 각각의 리소스를 사용하고, 상기 각각의 메모리, 상기 하드웨어 가속기 및 상기 호스트의 리소스들 사이에 포함되는 각각의 프로세서 사이에서 교환되는 데이터에 기초하여, 상기 각각의 작업 사양에 대한 명령들을 처리하는 동작을 포함하는, 시스템.

청구항 14

제13 항에 있어서,

상기 ML 워크로드를 수행하는 동작은,

상기 호스트의 각각의 리소스를 연결하는 하드웨어 소켓을 통해 교환되는 데이터에 기초하여 상기 명령들을 처리하는 것에 응답하여, 상기 각각의 작업 사양에 명시되는 작업들을 실행하는 동작을 포함하고,

상기 하드웨어 소켓은 상기 호스트에 의해 관리되는 다수의 리소스들 사이에서 공유되는 로컬 통신 버스를 규정하는, 시스템.

청구항 15

제13 항에 있어서,

제1 호스트에 대한 각각의 NUMA 토플로지는,

- i) 상기 제1 호스트에 로컬적인 리소스들의 각각의 구성의 각각의 제1 메모리; 및
- ii) 제2 다른 호스트에 로컬적이지만 상기 제1 호스트에 원격인 리소스들의 각각의 구성의 각각의 제2 다른 메모리에 부분적으로 기초하는, 시스템.

청구항 16

제11 항에 있어서,

상기 호스트들의 수량을 결정하는 동작은,

상기 복수의 호스트들의 각각의 호스트에 의해 관리되는 리소스들의 구성을 설명하는 시스템 파일을 취득하는 동작; 및

상기 복수의 호스트들의 각각의 호스트에 대한 시스템 파일에 설명되는 상기 리소스들의 구성에 기초하여 상기 호스트들의 수량을 결정하는 동작을 포함하는, 시스템.

청구항 17

삭제

청구항 18

제10 항에 있어서,

상기 호스트의 제어 그룹의 가속기들에 대한 하나 이상의 소켓 인터페이스들에 기초하여, 상기 작업 사양의 ML 작업을 상기 제어 그룹에 배정하는 동작 – 상기 소켓 인터페이스들은 상기 시스템 파일에 설명되는 NUMA 소켓들의 맵핑에 포함됨 –; 및

상기 제어 그룹의 가속기들을 사용하여 상기 제어 그룹 하의 공정으로서 상기 ML 작업을 실행하는 동작을 포함하는, 시스템.

청구항 19

복수의 신경망 레이어들을 포함하는 신경망을 구현하도록 각각 구성되는 하드웨어 가속기들을 사용하여, 기계-학습 워크로드를 수행하기 위해 작업들을 스케줄링하고 리소스들을 할당하기 위한 명령들을 저장하는 비일시적 기계 편독 가능한 저장 매체로서,

상기 명령들은 동작들의 수행을 야기하도록 하나 이상의 프로세서들에 의해 실행 가능하고,

상기 동작들은,

기계-학습(ML) 워크로드를 수행하라는 요청을 수신하는 동작;

상기 요청에 기초하여, 복수의 호스트들을 포함하는 분산 처리 시스템에서 상기 ML 워크로드를 수행하기 위해 리소스 요건을 결정하는 동작 – 상기 복수의 호스트들의 각각의 호스트는 각각의 복수의 하드웨어 가속기들을 포함함 –;

상기 각각의 호스트에 대한 각각의 복수의 하드웨어 가속기들 및 상기 리소스 요건에 기초하여, 상기 ML 워크로드를 형성하는 작업들의 세트로부터의 각각의 작업을 각각 실행하도록 배정되는 호스트들의 수량을 결정하는 동

작;

상기 수량의 호스트들의 각각의 호스트에 대해:

상기 복수의 호스트들의 각각의 호스트에 대한 불균일 메모리 액세스(NUMA) 소켓들의 맵핑을 설명하는 시스템 파일에 기초하여, 상기 호스트의 리소스들을 결합하는 하나 이상의 소켓들을 식별하는 동작;

상기 호스트의 리소스들을 결합하는 상기 하나 이상의 소켓들에 기초하여, 상기 호스트의 제어 그룹을 형성하는 동작;

상기 호스트의 메모리 액세스 토폴로지에 기초하여, 상기 각각의 복수의 하드웨어 가속기들을 포함하는 상기 호스트의 리소스들을 사용하여, 상기 호스트에서 실행되도록 배정된 작업을 명시하는 각각의 작업 사양을 생성하는 동작; 및

상기 수량의 호스트들의 상기 호스트에 상기 각각의 작업 사양을 제공하는 동작; 및

상기 수량의 호스트들의 각각의 호스트에 의해, 상기 호스트에 대한 각각의 작업 사양에 명시된 작업을 실행함으로써 상기 ML 워크로드를 수행하는 동작을 포함하는, 기계 판독 가능한 저장 매체.

청구항 20

제19 항에 있어서,

상기 각각의 호스트의 메모리 액세스 토폴리지는 상기 호스트에 로컬적인 각각의 메모리를 포함하는 각각의 NUMA 토폴리지를 포함하고, 그리고

상기 각각의 메모리는 상기 각각의 메모리를, 상기 각각의 복수의 하드웨어 가속기들 중 각각의 하드웨어 가속기 및 상기 호스트의 하나 이상의 다른 리소스들에 결합시키는 소켓 인터페이스를 포함하는, 기계 판독 가능한 저장 매체.

발명의 설명

기술 분야

[0001] 본 명세서는 일반적으로 컴퓨팅 워크로드들의 작업들의 스케줄링 및 컴퓨팅 워크로드들의 작업들을 수행하는 데 사용되는 리소스들의 할당에 관한 것이다.

배경 기술

[0002] 분산 컴퓨팅 시스템들에는 일반적으로 중앙 처리 장치들(CPUs), 저장 구성요소들 및 이미지/음성 처리 가속기들, 비디오 트랜스코딩 가속기들 또는 신경망 프로세서들(예를 들어, 기계-학습(ML) 가속기들)과 같은 다양한 리소스들이 포함된다. 이들 리소스들은 ML 시스템 트레이닝을 위한 워크로드 또는 이미지 분류나 음성 인식을 위한 전사(transcription) 생성을 위한 추론 워크로드와 같은 예시의 컴퓨팅 워크로드의 작업들을 처리하도록 상호작용할 수 있다.

[0003] 워크로드들을 처리하기 위한 기존의 방식들은 주기적으로, 메모리에 접속하고, 서로에 대해 비-로컬(또는 원격)인 분산 시스템의 컴퓨팅 리소스들 또는 리소스들의 그룹들 사이의 데이터 통신들을 교환할 필요가 있다. 이러한 비-로컬 메모리 접속 동작들 및 데이터 통신들은 종종 대역폭을 많이 사용하므로, 호스트의 크로스 소켓(예를 들어, 원격) 동작들에 대한 대역폭이 제한될 때 컴퓨팅 클러스터에서 성능 병목 현상이 발생할 수 있다.

발명의 내용

[0004] 본 문헌은 대규모 분산 시스템에 있어서 다수의 호스트들에 걸쳐 리소스들의 각각의 그룹들에 워크로드의 작업들을 배정함으로써, 기계-학습(ML) 워크로드 처리 시의 스케줄링 및 리소스 할당을 개선하기 위한 기술들을 설명한다. 본 문헌에 설명되는 기술들을 사용하여, 분산 시스템은 분산 시스템의 공유 또는 공통 하드웨어 버스를 통해 데이터 통신들을 교환하는 리소스들의 그룹에 워크로드의 각각의 작업을 배정하도록 구성될 수 있다. 이러한 배정 방식을 사용하면, 리소스 그룹의 불균일 메모리 액세스(NUMA) 토폴로지에 기초한 리소스 로컬리티(locality)를 활용하여 워크로드 처리 시간을 줄일 수 있다. 일부 예들에서, 설명된 기술들은 분산 시스

템의 개별 텐서 처리 노드들에서 수행되는 신경망 컴퓨테이션들을 가속화하기 위해 일단의 하드웨어 가속기들에 대한 NUMA 인식 스케줄링을 수행하는 데 사용될 수 있다.

[0005] 본 명세서에서 설명되는 청구 대상(subject matter)의 일 양태는, 다수의 신경망 레이어들을 포함하는 신경망을 구현하도록 각각 구성되는 하드웨어 가속기들을 사용하여 기계-학습 워크로드를 수행하기 위해 작업들을 스케줄링하고 리소스들을 할당하기 위한 방법으로 실현될 수 있다. 그 방법은, 기계-학습(ML) 워크로드를 수행하라는 요청을 수신하는 단계; 요청에 기초하여, 복수의 호스트들을 포함하는 분산 처리 시스템에서 ML 워크로드를 수행하기 위해 리소스 요건을 결정하는 단계 – 복수의 호스트들의 각각의 호스트는 각각의 복수의 하드웨어 가속기들을 포함함 –; 각각의 호스트에 대한 각각의 복수의 하드웨어 가속기들 및 리소스 요건에 기초하여, ML 워크로드를 형성하는 작업들의 세트로부터의 각각의 작업을 실행하도록 각각 배정되는 호스트들의 수량을 결정하는 단계를 포함한다.

[0006] 그 방법은, 그 수량의 호스트들의 각각의 호스트에 대해, 호스트의 메모리 액세스 토폴로지에 기초하여, 각각의 복수의 하드웨어 가속기들을 포함하는 호스트의 리소스들을 사용하여, 호스트에서 실행되도록 배정된 작업을 명시하는 각각의 작업 사양을 생성하는 단계와 그 수량의 호스트들의 호스트에 각각의 작업 사양을 제공하는 단계; 및 그 수량의 호스트들의 각각의 호스트에 의해, 호스트에 대한 각각의 작업 사양에 명시된 작업을 실행함으로써 ML 워크로드를 수행하는 단계를 포함한다.

[0007] 이들 및 다른 구현예들은 각각 선택적으로 이하의 특징들 중 하나 이상을 포함할 수 있다. 예를 들어, 일부 구현예들에 있어서, 각각의 호스트의 메모리 액세스 토폴로지는 호스트에 대해 로컬적인 각각의 메모리를 포함하는 각각의 불균일 메모리 액세스(NUMA) 토폴로지를 포함하고; 그리고 각각의 메모리는 각각의 메모리를, 각각의 복수의 하드웨어 가속기들 중 각각의 하드웨어 가속기 및 호스트의 하나 이상의 다른 리소스들에 결합시키는 소켓 인터페이스를 포함한다.

[0008] 일부 구현예들에 있어서, 각각의 작업 사양에 명시된 작업을 실행하는 것은, 다수의 신경망 컴퓨테이션들의 각각의 부분들을 각각의 복수의 하드웨어 가속기들의 각각의 하드웨어 가속기에 배정하는 것에 응답하여, 복수의 신경망 레이어들 중 각각의 신경망 레이어에 대한 출력을 생성하도록 다수의 신경망 컴퓨테이션들을 수행하는 것을 포함한다.

[0009] 일부 구현예들에 있어서, ML 워크로드를 수행하는 단계는, 호스트의 제어 그룹의 각각의 리소스를 사용하고, 각각의 메모리, 하드웨어 가속기 및 호스트의 리소스들 사이에 포함되는 각각의 프로세서 사이에서 교환되는 데이터에 기초하여, 각각의 작업 사양에 대한 명령들을 처리하는 단계를 포함한다.

[0010] 일부 구현예들에 있어서, ML 워크로드를 수행하는 단계는, 호스트의 제어 그룹의 각각의 리소스를 연결하는 하드웨어 소켓을 통해 교환되는 데이터에 기초하여 명령들을 처리하는 것에 응답하여, 각각의 작업 사양에 명시되는 작업들을 실행하는 단계를 포함하고, 하드웨어 소켓은 호스트에 의해 관리되는 다수의 리소스들 사이에서 공유되는 로컬 통신 버스를 규정한다.

[0011] 일부 구현예들에 있어서, 제1 호스트에 대한 각각의 NUMA 토폴로지는, i) 제1 호스트에 로컬적인 리소스들 각각의 구성의 각각의 제1 메모리; 및 ii) 제2 다른 호스트에 로컬적이지만 제1 호스트에 원격인 리소스들 각각의 구성의 각각의 제2 다른 메모리에 부분적으로 기초한다.

[0012] 일부 구현예들에 있어서, 호스트들의 수량을 결정하는 단계는, 복수의 호스트들 각각의 호스트에 의해 관리되는 리소스들의 구성을 설명하는 시스템 파일을 취득하는 단계; 및 복수의 호스트들 각각의 호스트에 대한 시스템 파일에 설명되는 리소스들의 구성에 기초하여, 호스트들의 수량을 결정하는 단계를 포함한다. 일부 구현예들에 있어서, 그 방법은 복수의 호스트들 각각의 호스트에 대한 NUMA 소켓들의 맵핑을 설명하는 시스템 파일에 기초하여, 호스트의 리소스들을 결합하는 하나 이상의 소켓들을 식별하는 단계; 및 호스트의 리소스들을 결합하는 하나 이상의 소켓들에 기초하여, 호스트의 제어 그룹을 형성하는 단계를 포함한다.

[0013] 일부 구현예들에 있어서, 호스트의 제어 그룹의 가속기들에 대한 하나 이상의 소켓 인터페이스들에 기초하여, 작업 사양의 ML 작업을 제어 그룹에 배정하는 단계 – 소켓 인터페이스들은 시스템 파일에 설명되는 NUMA 소켓들의 맵핑에 포함됨 –; 및 제어 그룹의 가속기들을 사용하여, 제어 그룹 하의 공정으로서 ML 작업을 실행하는 단계를 포함한다.

[0014] 이러한 양태들 및 다른 양태들의 다른 구현예들은 대응하는 시스템들, 장치 및 컴퓨터 프로그램들을 포함하며, 이들은 비일시적 컴퓨터 판독 가능한 저장 디바이스들에 암호화되어 방법들의 거동들을 수행하도록 구성된다. 하나 이상의 컴퓨터들의 시스템은 작동시 시스템에 의해 그 거동들이 수행될 수 있도록 하는 시스템

상에 설치되는, 소프트웨어, 펌웨어, 하드웨어 또는 이들의 조합에 의해 그렇게 구성될 수 있다. 하나 이상의 컴퓨터 프로그램들은 데이터 처리 장치에 의해 실행될 때 그 장치가 거동들을 수행하게 하는 명령들을 가짐으로써 그렇게 구성될 수 있다.

[0015] 본 명세서에서 설명되는 청구 대상의 일 양태는, 워크로드들을 수행하라는 요청을 수신하고, 그 요청에 기초하여, 워크로드들을 수행하기 위해 리소스 요건들을 결정하는 시스템에서 실현될 수 있다. 그 시스템은 다수의 호스트들을 포함하고, 각각의 호스트는 다수의 가속기들을 포함한다. 그 시스템은 각각의 호스트에 대한 가속기들 및 리소스 요건에 기초하여, 워크로드의 작업들을 실행하도록 배정되는 호스트들의 수량을 결정한다. 그 수량의 호스트들 각각의 호스트에 대해, 그 시스템은 호스트의 메모리 액세스 토폴로지에 기초하여, 작업 사양을 생성한다. 사양은 다수의 가속기들을 포함하는 호스트의 리소스들을 사용하여 호스트에서 실행되는 작업을 명시한다. 그 시스템은 작업 사양을 호스트들에 제공하고, 각각의 호스트가 호스트에 대한 작업 사양에 명시된 배정된 작업들을 실행할 때 워크로드들을 수행한다.

[0016] [0016] 본 명세서에 설명된 청구 대상은 이하의 이점들 중 하나 이상을 실현하기 위해 특정 실시예들에서 구현될 수 있다. 본 문헌에서 설명되는 기술들은 시스템의 호스트들이 워크로드의 작업들을 실행할 때, 비-로컬 메모리 액세스 동작들 및 데이터 통신들의 발생을 감소시키거나 방지함으로써, 시스템 내 성능 병목 현상들을 완화할 수 있다. 종래 접근법들에 비해, 설명되는 기술들은 시스템 각각의 호스트에 의해 관리되는 리소스들의 그룹들 또는 리소스들의 불균일 메모리 액세스(NUMA) 토폴로지에 기초하는 리소스 로컬리티를 활용함으로써, 워크로드들을 처리하는 데 요구되는 시간량을 감소시키는 데 사용될 수 있다.

[0017] [0017] 본 명세서에 설명되는 청구 대상의 하나 이상의 구현예들의 상세가 첨부 도면들 및 이하의 설명에 개시된다. 청구 대상의 다른 잠재적인 특징들, 양태들 및 이점들은 발명의 설명, 도면들 및 청구범위로부터 명백할 것이다.

도면의 간단한 설명

[0018] [0018] 도 1은 기계-학습 워크로드들을 수행하기 위해 실행되는 작업들을 스케줄링하기 위한 예시의 컴퓨팅 시스템의 블록도이다.

[0019] [0019] 도 2는 도 1의 컴퓨팅 시스템에 포함되는 호스트에 의해 관리되는 예시의 리소스들의 블록도를 도시한다.

[0020] [0020] 도 3은 기계-학습 워크로드를 수행하기 위한 작업 사양을 생성하기 위해 실행될 수 있는 예시의 컴퓨팅 로직을 도시한다.

[0021] [0021] 도 4는 기계-학습 워크로드들을 수행하기 위해 실행되는 작업들을 스케줄링하기 위한 예시의 공정을 도시한다.

[0022] [0022] 도 5는 도 1의 컴퓨팅 시스템의 호스트에 제공되는 작업 사양을 생성하기 위한 예시의 공정을 도시한다.

[0023] [0023] 다양한 도면들에 있어서 유사한 참조번호들 및 명칭들은 유사한 요소들을 표시한다.

발명을 실시하기 위한 구체적인 내용

[0024] [0024] 분산 시스템들은 컴퓨팅 워크로드를 수행하는 데 사용되는 하드웨어 디바이스들을 포함하는 다수의 노드들(예를 들어, 호스트들)을 구비한다. 노드들은 워크로드를 수행하기 위해 데이터를 처리하는, 개별적인 하드웨어 컴퓨팅 클러스터들 또는 클러스터 내 호스트 디바이스들을 형성할 수 있다. 노드들 각각은 다수의 리소스들을 포함할 수 있다. 예를 들어, 리소스는 프로세서나 CPU, 메모리 또는 주변 부품 상호연결(PCI) 디바이스(예를 들어, 하드웨어 가속기)일 수 있으며, 각각의 호스트는 리소스 그룹을 형성하는 다수의 리소스들을 포함할 수 있다.

[0025] [0025] 각각의 리소스는 일부 리소스들이 노드 내 다른 리소스들에 대해 원격 또는 로컬적이도록 하는 소정의 하드웨어 또는 소켓 연결을 가질 수 있다. 분산 시스템에서 워크로드들을 처리하기 위한 접근법들은 메모리에의 접속 및 서로에 대해 비-로컬(또는 원격)인 리소스들 사이에서의 데이터 이동을 포함하는 동작들의 수행을 필요로 하는 경우가 종종 있다. 전술한 바와 같이, 이러한 비-로컬 메모리 접속 및 데이터 전송 동작들은 크로스 소켓(예를 들어, 원격) 동작들을 위한 대역폭을 제한한 클러스터 또는 호스트에 있어서 성능 병목 현상들을 야기할 수 있다.

- [0021] [0026] 이러한 맥락에서, 분산 시스템의 컴퓨테이셔널 클러스터들에서 컴퓨팅 워크로드들의 작업들을 스케줄링하고, 컴퓨팅 워크로드들을 형성하는 작업들을 수행하는 데 사용되는 시스템의 리소스들을 할당함에 있어서의 개선을 위한 기술들이 설명된다. 스케줄링과 관련하여, 그 기술들은 대규모 분산 시스템의 개별 호스트들에 의해 관리되는 리소스들의 각각의 그룹들로 워크로드(예를 들어, ML 워크로드)의 작업들을 배정하기 위한 개선된 공정을 포함한다. 예를 들어, 그 시스템은 리소스들의 특정 그룹으로 워크로드의 구체적인 작업들을 배정하도록 구성되며, 그룹 내 개별 리소스들은 분산 시스템의 공유 또는 공통의 하드웨어 버스를 통해 데이터 통신들을 교환한다. 소정의 작업들을 호스트의 특정 리소스 그룹들에 배정하기 위한 공정은 컴퓨팅 클러스터 내 리소스로 커리티를 활용하여 수행되며, 로컬리티는 리소스 그룹의 불균일 메모리 액세스(NUMA) 토플로지에 기초한다.
- [0022] [0027] 분산 시스템의 별개의 컴퓨팅 클러스터들 내 다수의 호스트들에 걸쳐 NUMA 인식 작업 스케줄링 및 리소스 할당을 수행하기 위한 접근법을 또한 제공하는 기술들이 설명된다. 예를 들어, 컴퓨팅 클러스터의 제어기는 NUMA 로컬리티를 요구하는 워크로드의 작업들 또는 과제들을 설명하는 프로토콜 비트들의 세트를 전달할 수 있다. 그 기술들은 CPU, 메모리 및 주변 부품 상호연결(PCI) 디바이스(예를 들어, 하드웨어 가속기)를 포함하는 리소스 그룹과 같은 특정 리소스들 또는 디바이스들의 세트에 하나 이상의 작업들을 배정함으로써, 클러스터 또는 클러스터 내의 호스트의 NUMA 로컬리티를 활용한다.
- [0023] [0028] 다수의 리소스 그룹들을 관리하는 호스트는 클러스터의 마스터 제어기에 의해 전달된 프로토콜 비트들을 수신하여 처리하도록 작동 가능하다. 예를 들어, 호스트는 호스트의 리소스 그룹들의 결정된 소켓 토플로지에 기초하여 프로토콜 비트들을 처리하고, 동일한 NUMA 소켓으로부터의 특정 리소스들의 세트를 할당하여 프로토콜 비트들에 의해 명시된 것과 같은 워크로드의 특정 과제 또는 부분에 대한 작업들을 실행한다. 호스트는 작업 사양의 기계-학습 작업들의 세트를 호스트에서 구성된 리소스 그룹(또는 제어 그룹)에 결합 또는 배정하도록 작동 가능하다. 예를 들어, 프로토콜 비트들에 의해 전달된 정보에 기초하여, 호스트는 주어진 리소스 그룹에 작업을 결합하여, 주어진 작업에 대한 컴퓨테이션들의 성능 또는 실행을 저하시킬 수 있는 비-로컬 메모리 또는 데이터 접속 동작들의 발생을 감소 또는 방지할 수 있다.
- [0024] [0029] NUMA 인식 작업 스케줄링 및 리소스 할당에 대해 설명된 접근법을 통해 분산 시스템은 그것의 NUMA 로컬리티의 사용을 최적화하여 대역폭 요건들을 감소시키고 소정의 워크로드들을 수행하기 위한 컴퓨팅 시간들을 개선할 수 있다. 예를 들어, 이것은 적어도 클러스터의 NUMA 로컬리티의 활용을 기초로 하여 달성될 수 있으며, 이로 인해 소정 유형들의 작업들이 크로스 소켓 또는 크로스 노드 통신들을 줄여 다른 컴퓨테이션들에 대한 대역폭을 확보할 수 있도록 병치되는 특정 디바이스들 또는 리소스 그룹들의 세트에 배정될 수 있다.
- [0025] [0030] 도 1은 컴퓨팅 워크로드들을 수행하도록 실행되는 작업들을 스케줄링하기 위한 예시의 분산 컴퓨팅 시스템(100)의 블록도이다. 시스템(100)은 다수의 컴퓨팅 클러스터들(102)을 포함하는 대규모 분산 하드웨어 컴퓨팅 시스템일 수 있으며, 각각의 클러스터(102)는 다수의 호스트들(104)을 포함하고, 각각의 호스트(104)는 다수의 컴퓨팅 리소스들(105)을 포함한다.
- [0026] [0031] 리소스들의 하나 이상의 그룹들은 분산 시스템(100)의 호스트(104)에 의해 관리될 수 있으며, 다수의 컴퓨팅 클러스터들(102) 각각은 다수의 호스트들(104)을 포함할 수 있다. 더욱 구체적으로, 각각의 호스트(104)는 리소스들의 그룹을 형성하는 2 개 이상의 개별 리소스들(105)을 관리하도록 구성된다. 호스트(104)에 의해 관리되는 리소스들의 그룹은 본 명세서에서는 대안적으로 리소스 그룹으로 언급될 수 있다. 그래서, 일부 케이스들에 있어서, 리소스(105)는 단일 프로세서 또는 메모리 디바이스와 같은 개별 리소스를 나타낼 수 있는 반면, 다른 케이스들에서는, 리소스(105)는 2 개 이상의 프로세서들, 2 개 이상의 메모리 뱅크들, 2 개 이상의 하드웨어 가속기들, 또는 그 각각의 결합들과 같은 다수의 리소스들을 나타낼 수 있다. 호스트(104)의 리소스 그룹들은 도 2를 참고하여 이하 더욱 상세하게 설명된다.
- [0027] [0032] 일부 구현예들에 있어서, 호스트(104)는 하드웨어-컴퓨팅 디바이스(예를 들어, 컴퓨터 또는 서버)이다. 일부 구현예들에서, 호스트(104)는 분산 시스템(또는 컴퓨팅 클러스터)의 가상 기계, 컴퓨팅 리소스들(105)의 그룹을 관리하기 위한 소프트웨어 구성, 또는 이 둘 모두이다. 시스템(100)은 M 개의 컴퓨팅 클러스터들(102)을 포함할 수 있고, M 개의 컴퓨팅 클러스터들 각각의 컴퓨팅 클러스터(102)는 N 개의 호스트들을 포함할 수 있으며, M 및 N 각각은 1 이상인 정수이다.
- [0028] [0033] 일부 구현예들에 있어서, 컴퓨팅 클러스터들(102) 각각은 클러스터(102)의 호스트들(104)을 형성하는 기계들(예를 들어, 하드웨어 또는 가상 기계들)의 세트를 포함한다. 도 1에 도시된 바와 같이, 단일 클러스터(102)는 다수의 제어기들(108)을 포함할 수 있으며, 각각의 제어기는 클러스터(102)에서 호스트들(104) 중 하나

이상에 워크로드의 작업들을 배정하는 기능을 한다.

[0029]

[0034] 컴퓨팅 클러스터들(102) 각각은 마스터 제어기(108)에 의해 접속 가능한 링크 샤드(link shard)(110) 뿐만 아니라, 클러스터(102)의 마스터 제어기(108)("제어기(108)")와 통신하는 스케줄러(106)를 포함할 수 있다. 제어기(108)는 작업 사양들의 생성과 호스트들(104)에 발송하기 위한 명령들 및 지시들의 준비, 및 호스트(104)로부터의 응답들에 기초한 호스트(104)의 현재 처리 상태의 업데이트를 전달한다. 일부 구현예들에 있어서, 각각의 제어기(108)는 호스트들(104)의 서브세트들과의 통신들을 관리하는 상태 로직(110)을 포함한다. 예를 들어, 상태 로직(110)은 호스트(104)의 처리 상태에 대한 정보를 취득하기 위해 호스트들(104)의 서브세트로 지시들을 발송하고 호스트들(104)로부터의 응답들을 수신하도록, 제어기(108)에 의해 운용된다. 예를 들어, 상태 로직(110)은 배정된 작업이 완료되었는지 또는 진행 중인지를 나타내는 호스트 레포트들을 수신 및 처리하는 데 사용된다. 상태 로직(110)은 호스트의 처리 상태에 대한 정보를 취득하고자 임계 횟수의 시도들 후, 호스트(104)가 상태 레포트를 제공하지 못하는 경우 호스트(104)에 배정된 작업이 정지된 것으로 결정할 수 있다. 일부 케이스들에 있어서, 상태 로직(110)은 마스터 제어기(108)에서 수신된 업데이트 로드들의 크기를 감소시키기 위해, 호스트들(104)에 의해 보고된 처리 상태 정보를 집계 및 압축하도록 작동될 수 있다.

[0030]

[0035] 이하 더욱 상세하게 설명되는 바와 같이, 스케줄러(106) 및 제어기(108)는 호스트에서의 실행을 위해, 워크로드의 작업들을 스케줄링하고 특정 호스트(104)로 배정하도록 상호작용하거나 통신한다. 비록, 도 1에서 제어기(108)와 분리되어 있는 것으로 도시되어 있지만, 스케줄러(106)는 제어기(108)에 통합될 수 있다. 일부 구현예들에 있어서, 스케줄러(106)는 컴퓨팅 클러스터들(102)의 선택적인 처리 요소이며, 스케줄러의 기능들은 제어기(108)에 구성되는 배정 및 제어 기능들에 통합될 수 있다.

[0031]

[0036] 제어기들(108)은 적어도 요청(112)을 기초로 워크로드의 작업들을 배정하여 호스트(104)에 의해 관리되는 리소스들의 하드웨어 구성 및 하나 이상의 워크로드들을 수행하도록 작동될 수 있다. 예를 들어, 제어기들(108) 각각은 클러스터(102)에서 수신된 요청(112)에 있어서의 매개변수들에 기초하여, 그리고 호스트(104)에 포함되는 리소스들(105)(또는 리소스 그룹들)의 하드웨어 소켓 토플로지에 기초하여, 명령들을 생성하는 논리적으로 중앙 집중화된 제어기일 수 있다. 일부 구현예들에 있어서, 호스트들(104)의 서브세트에 있어서의 각각의 호스트(104)는 특정 마스터 제어기(108) 하에 있는 "슬레이브" 컴퓨팅 자산들로서 구성된다. 이 구현예들에 있어서, 마스터 제어기(108)는 요청(112) 내 매개변수들 및 특정 마스터 제어기(108) 하의 "슬레이브"인 호스트들(104)의 서브세트의 각각의 호스트(104)에서의 하드웨어 소켓 토플로지들에 기초하여 명령들을 생성한다.

[0032]

[0037] 호스트(104)는 기계들 또는 하드웨어 디바이스들, 예를 들어 수백 또는 수 천의 리소스들 또는 디바이스들에 대응하는 다수의 리소스들(105)을 포함할 수 있다. 호스트(104) 내 리소스들(105)은 여러 측면들에 있어서 다양하거나 이종일 수 있다. 예를 들어, 호스트(104)에 의해 관리되는 리소스들(105)의 각각의 그룹은 처리 디바이스들(예를 들어, CPU, RAM, 디스크, 네트워크), 프로세서 유형, 처리 속도, 성능 및 외부 IP 주소 또는 플래시 저장과 같은 역량들의 면에서 다양할 수 있다. 더욱 구체적으로, 각각의 컴퓨팅 클러스터(102)에 대해, 클러스터(102) 내 다수의 호스트들(104)의 각각의 호스트는 워크로드의 작업들을 실행하기 위해 호스트의 다른 리소스들과 상호작용하는 하나 이상의 특수 목적 하드웨어 회로들을 포함한다.

[0033]

[0038] 예를 들어, 특수 목적 하드웨어 회로들은 하드웨어 가속기들, 그래픽 처리 장치(GPU) 하드웨어 가속기들 또는 신경망 프로세서들일 수 있다. 도 1의 예에서, 시스템(100)은 제1 호스트(104-1), 제2 호스트(104-2), 제3 호스트(104-3) 및 N 개의 추가 호스트들(104-n)을 포함할 수 있다. 일부 구현예들에 있어서, 제1 호스트(104-1)의 특수 목적 회로들 및 리소스들(105)은 제2 호스트(104-2)의 특수 목적 회로들 및 리소스들(105)과는 (예를 들어, 약간 또는 현저하게) 다를 수 있다.

[0034]

[0039] 예를 들어, 제1 호스트(104-1)는, 각각이 위치 기반 및 메모리 내 분석 또는 GPU 가속 데이터베이스 쿼리(query)들을 수행하도록 구성된 10 개의 GPU 하드웨어 가속기들을 포함할 수 있는 반면, 제2 호스트(104-2)는, 각각이 컨볼루션 신경망(Convolutional Neural Network; CNN) 모델 또는 순환 신경망(recurrent neural network; RNN) 모델을 구현하도록 구성된 20 개의 신경망 프로세서들을 포함할 수 있다. 일부 구현예들에 있어서, 20 개의 신경망 프로세서들은 트레이닝된 추론 모델들에 대해 이진법을 실행하고, 부동소수점 기반 추론 모델, 정수 양자화 추론 모델 또는 둘 모두의 운용을 가속화하도록 구성될 수 있다.

[0035]

[0040] 시스템(100)은 제어기들(108)을 사용하여, 특정 호스트(104)의 하나 이상의 기계들에서 운용될 수 있는 작업들과 같은 개별적인 작업들의 실행을 배정 및 제어하기 위한 명령들을 생성한다. 호스트(104)의 특정 리소스 그룹들에 소정의 작업들을 배정하기 위한 결정들은 컴퓨팅 클러스터(102)의 호스트(104) 내에서 리소스 로컬리티를 활용하는 데 특히 중점을 두고 형성된다. 리소스 로컬리티는 호스트(104)에서의 리소스 그룹들의 하드

웨어 토플로지에, 그리고 더욱 구체적으로는 후술하는 바와 같이, 리소스 그룹의 불균일 메모리 액세스(NUMA) 토플로지에 기초한다.

- [0036] [0041] 시스템(100)은 각각의 컴퓨팅 클러스터(102) 및 컴퓨팅 클러스터의 각각의 호스트(104)에 대한 하드웨어 토플로지들을 포함하는 시스템 토플로지를 생성하도록 구성된다. 하드웨어 토플로지는 i) 호스트의 다수의 디바이스들 및 리소스들의 연결(예를 들어, 소켓 연결부들 및 인터페이스들) 및 ii) 호스트(104)의 리소스들(105) 사이의 데이터 전송들을 가능하게 하는 로컬 통신 버스들을 식별하도록 구성된다.
- [0037] [0042] 시스템(100)은 호스트(104)에서 하드웨어 소켓의 연결 지점 또는 구성요소 인터페이스에 결합된 각각의 리소스 또는 주변 디바이스의 위치들을 식별하도록 구성된다. 예를 들어, 호스트(104)는 호스트(104)에 의해 관리되는 하드웨어 컴퓨터의 시스템 BIOS와 연관된 프로그램 코드(예를 들어, 펌웨어)를 운용하여 컴퓨터의 마더보드에 결합된 리소스들(105)(예를 들어, 프로세서들 및 메모리)의 유형들과 리소스 위치들을 식별할 수 있다. 일부 구현예들에 있어서, 호스트(104)의 동작 시스템은 하드웨어 컴퓨터의 칩셋을 사용하여 호스트(104)에 의해 관리되는 컴퓨터에 연결된 데이터 버스들과 주변 디바이스들에 대한 정보의 상세 목록을 취득할 수 있다. 예를 들어, 그 목록은 컴퓨터의 프로세서에서 운용되는 동작 시스템의 상호연결 구성 공간을 나타내는 일반 휴대용 상호연결 라이브러리(예를 들어, libpci)에 기초할 수 있다.
- [0038] [0043] 제어기(108)는 호스트(104)에서 관리 중인 특정 기계 또는 리소스 그룹(105)의 현재 처리 상태에 대한 정보를 취득하기 위해, 각각의 호스트(104)로 지시들을 전송할 뿐만 아니라, 작업을 실행하기 위해 각각의 호스트(104)로 미처리 명령들을 전송하도록 작동할 수 있다. 일부 구현예들에 있어서, 제어기(108)는 처리 상태에 대한 정보를 취득하기 위해 지시들을 동적으로 전송한다. 대안적으로, 제어기(108)는 미리 결정된 스케줄(예를 들어, 몇 초마다)을 참조하여 처리 상태에 대한 정보를 취득하기 위해 지시를 전송할 수 있으며, 그 스케줄은 호스트(104)에서 실행 중인 특정 작업에 기초한다. 일반적으로, 각각의 제어기(108)는 제어기가 호스트로 전송하는 명령들 및 지시들에 기초하여, 호스트(104)의 다양한 리소스들과 다른 리소스 그룹들 사이에서의 각각의 통신 속도들을 제어하도록 작동 가능하다.
- [0039] [0044] 도 2는 예시의 컴퓨팅 클러스터(102)의 호스트(104)에 의해 관리되는 예시의 리소스 그룹들(200)의 블록도를 도시한다. 전술한 바와 같이, 호스트(104)는 기계들 또는 하드웨어 디바이스들에 대응하는 수백 또는 수천의 리소스들을 포함할 수 있다. 호스트(104)의 리소스 그룹 내 리소스들(105)은 여러 측면들에서 다양하거나 이종일 수 있다. 예를 들어, 호스트(104)에 의해 관리되는 리소스들(200) 각각의 그룹은 처리 디바이스들(예를 들어, CPU, RAM, 디스크, 네트워크), 프로세서 유형, 처리 속도, 전체 성능 및 외부 IP 주소 또는 플래시 저장 장치와 같은 역량들의 면에서 다양할 수 있다.
- [0040] [0045] 도 2에 도시된 바와 같이, 각각의 호스트(104)의 메모리 액세스 토플리지들은 호스트(104)에 의해 관리되는 하나 이상의 리소스 그룹들(200) 각각의 불균일 메모리 액세스(NUMA) 토플리지 또는 소켓(202-1, 202-2)을 포함할 수 있다. 리소스 그룹(200)의 NUMA 토플리지는 다수의 프로세서들(P)(204) 또는 다수의 프로세서 코어들(P), 메모리 리소스, 예를 들어 랜덤 액세스 메모리(RAM) 및 하드웨어 가속기(208)와 같은 하나 이상의 특수 목적 회로들을 포함할 수 있다. NUMA 토플리지의 개별적인 리소스들은 NUMA 소켓(202-1 또는 202-2)에 대응하는 로컬 NUMA 노드를 형성할 수 있다.
- [0041] [0046] 예를 들어, 로컬 NUMA 노드는 공유 또는 공통 하드웨어 버스(210)를 통해 데이터 통신들을 교환하는 그룹 내 리소스들에 기초하여 형성될 수 있다. 로컬 NUMA 노드 내 각각의 리소스는 리소스들이 공통 소켓에의 인터페이스(또는 소켓) 연결을 통해 노드에 연결될 때 다른 리소스에 대해 로컬일 수 있다. 일부 구현예들에 있어서, 각각의 하드웨어 가속기(208)는 PCI 또는 PCI-e 소켓 연결을 통해 NUMA 노드의 다른 리소스들에 연결된다.
- [0042] [0047] 본 명세서에서 사용된 바와 같이, NUMA는 분산 다중 처리 시스템들에서 사용되는 컴퓨터 메모리 디자인에 관한 것이며, 메모리 접속 시간은 프로세서(P)(204) 또는 프로세서 코어들에 대한 메모리 위치에 좌우된다. NUMA 하에서, 프로세서(204)는 프로세서들 사이에 공유되는 메모리 또는 다른 프로세서에 대해 로컬적인 메모리(206-2)와 같은 비-로컬 메모리보다 빠른 그 자신의 로컬 메모리(206-1)에 접속할 수 있다.
- [0043] [0048] 예시의 리소스 그룹(202)은 다수의 상호연결 위치들(212)을 포함할 수 있다. 예를 들어, 상호연결 위치들(212-1 및 212-2) 각각은 호스트(104)의 메모리(206-1)와 하드웨어 가속기(208) 사이와 같은 호스트(104)의 리소스들(105) 사이의 데이터 연결들을 설정하기 위한 각각의 구성요소 인터페이스들에 대응할 수 있다. 일부 구현예들에 있어서, 리소스 그룹(200)의 리소스들(105)은 NUMA 소켓(202-1)의 로컬 리소스들을 연결하는 하드웨

어 소켓을 통해 데이터 통신들을 교환하며, 하드웨어 소켓은 호스트(104)에 의해 관리되는 다수의 리소스들 사이에서 공유되는 로컬 통신 버스(210)를 규정한다.

[0044] 일부 구현예들에 있어서, 제1 NUMA 소켓(202-1)에 대한 각각의 NUMA 토플리지는 i) NUMA 소켓(202-1)에 대해 로컬적인 리소스들 각각의 구성에 있어서의 각각의 제1 메모리(206-1) 및 ii) 제2 다른 NUMA 소켓(202-2)에 대해 로컬적이지만 제1 NUMA 소켓(202-1)에 대해 원격인 리소스들 각각의 구성에 있어서의 각각의 제2 다른 메모리(206-2)에 부분적으로 기초한다.

[0045] 도 3은 컴퓨팅 워크로드를 수행하기 위해 시스템(100)에서 실행되는 컴퓨팅 로직(302)에 기초하는 예시의 작업 사양(300)을 도시한다. 도 3에 도시된 바와 같이, 로직(302)은 각각이 명령들(예를 들어, 프로그램된 코드/명령들)을 포함하는 다수의 컴퓨팅 블록들을 포함할 수 있다. 명령들은 제어기(108)의 처리 디바이스들, 호스트(104)의 처리 디바이스들 및 다른 리소스들(105) 또는 각각의 조합들을 사용하여 시스템(100)에서 실행될 수 있다.

[0046] 컴퓨팅 로직(302)은 시스템(100)에서 ML 워크로드를 수행하기 위해 작업들을 스케줄링하고 리소스들을 할당하기 위한 예시의 작업 사양(300)의 프로그래밍 표현일 수 있다. 일부 구현예들에 있어서, ML 워크로드는 각각이 다수의 신경망 레이어들을 포함하는 신경망을 구현하도록 구성되는 하드웨어 가속기들을 사용하여 수행된다. 명령들은 시스템(100)의 하나 이상의 비일시적 기계 판독 가능한 저장 매체들에 저장될 수 있으며, 시스템(100)의 하나 이상의 프로세서들에 의해 실행되어 동작들의 수행을 야기하고 워크로드의 작업들을 실행할 수 있다.

[0047] 예를 들어, 동작들은 ML 워크로드의 작업들을 실행하기 위해 특정 호스트(104)에 제공되는 (예를 들어, 작업 사양을 위한) 명령들 및 프로토콜 비트들을 생성하도록 수행될 수 있다. 일부 구현예들에 있어서, 프로토콜 비트들은 코드화된 신호들, 이진수 값들의 데이터 워드 또는 다른 관련 매개변수들이나 데이터 값들에 의해 표시된다. 코드화된 신호들 및 이진수 값들은 작업 사양의 작업들의 배정을 결정하기 위해 호스트(104)에서 수신 및 처리(또는 해석)된다. 일반적으로, 각각의 호스트(104)는 호스트(104)에 배정될 수 있는 다수의 다른 워크로드들의 작업들을 포함한 워크로드의 하나 이상의 작업들을 운용 또는 실행하도록 구성된다. 요청(112)이 컴퓨팅 클러스터(102)에 의해 수신되는 경우, 클러스터(102)의 스케줄러(106) 및 제어기(108)는 요청(112)을 스캔하도록 상호작용한다.

[0048] 예를 들어, 제어기(108)는 워크로드에서 다양한 작업들의 CPU, 메모리 및 가속기 요건들을 명시하는 요청(112) 내 매개변수들(예를 들어, 프로토콜 비트들)을 식별하기 위해 요청(112)을 스캔할 수 있다(304). 요청(112) 내 매개변수들 및 값들에 기초하여, 제어기(108)는 예시의 워크로드가 16 개의 작업들을 포함한다고 결정할 수 있으며, 여기서 작업들 각각은 96 개의 CPU들 및 4 개의 특수 목적 회로들, 예를 들어 하드웨어 가속기들의 전체 리소스 할당을 요구한다. 예를 들어, 요청(112)은 16 개의 작업들 각각을 실행하기 위해 사용될 하드웨어 가속기들(4)의 수량을 명시하는 스칼라 리소스 매개변수를 포함할 수 있다. 일부 구현예들에 있어서, 스칼라 리소스 매개변수는 워크로드를 처리하는 데 사용되는 하드웨어 가속기의 유형을 명시하는 하부-유형을 포함할 수 있다. 예를 들어, 하부-유형은 4 개의 하드웨어 가속기들 각각이 특정부 인식을 위해 트레이닝된 모델의 운용을 가속화하도록 구성된 신경망 프로세서인 것으로 명시할 수 있다.

[0049] 컴퓨팅 로직의 패키지 필드는 16 개의 작업들 각각을 실행하기 위한 작업 바이너리를 명시한다(306). 예를 들어, 작업 바이너리는 16 개의 작업들 중 특정 작업을 실행하기 위한 컴퓨테이션들을 수행하기 위해 하드웨어 가속기에서 실행 또는 운용될 특정 유형의 신경망 또는 추론 모델일 수 있다. 일부 케이스들에 있어서, 작업 바이너리는 워크로드의 작업들을 처리하는 데 사용되는 하드웨어 가속기의 유형을 명시하는 스칼라 리소스 하부-유형으로부터 유래된다.

[0050] 요청(112)에서의 매개변수들의 식별에 응답하여, 제어기(108)는 요청(112)의 매개변수들에 기초하여, 그리고 호스트(104) 내 리소스들(105)(또는 리소스 그룹들(200))의 하드웨어 소켓 토플로지에 기초하여, 작업들의 스케줄링 및 클러스터(102)의 호스트들(104)로의 배정을 위한 배정 방식을 결정하도록 작동 가능하다. 제어기(108)는 작업들의 스케줄링 및 호스트들(104)로의 배정을 위한 배정 방식에 기초하여 각각의 작업 사양을 생성한다. 예를 들어, 요청(112) 내 각각의 매개변수들 및 대응하는 매개변수 값들은 스케줄러(108)에 대한 스케줄링 제한을 나타낼 수 있다. 일부 구현예들에 있어서, 요청(112)은 스케줄러(108) 및 제어기(108)를 추가로 제한하기 위해 매개변수들 각각에 대한 우선순위를 배정할 수 있다. 예를 들어, 가속기 서브-유형들 또는 CPU 코어들에 배정된 우선순위들은 특정 유형들의 하드웨어 가속기들 또는 특정 수량의 사용 가능한 CPU들을 갖는 소정의 호스트들(104)에 대해 제어기(108)를 제한할 수 있다.

[0051]

[0056] 제어기(108)는 적어도, 제어기(108) 하의 "슬레이브들"인 호스트(104) 중 하나 이상에 의해 관리되는 리소스 그룹들 각각에 대한 하드웨어 소켓 토폴로지의 상세들에 대해 요청(112)의 매개변수들을 분석함으로써, 작업들의 배정을 결정하고, 작업 사양들을 생성한다. 예를 들어, 제어기(108)는 리소스들(105)의 로컬리티를 결정하고, 리소스들 또는 리소스들의 유형들이 요청(112)의 제한들을 만족하는지 여부를 결정하고, 그리고 리소스들의 사용 가능성을 결정하기 위해, 호스트(104)의 각각의 리소스 그룹의 하드웨어 소켓 토폴로지를 스캔하도록 작동 가능하다. 일부 예들에 있어서, 제어기(108)는 특정 NUMA 노드에 대해 로컬적이고, 요청(112)의 제한들 중 하나 이상을 만족하는 리소스들 중으로부터 리소스들의 사용 가능성을 결정한다. 일부 구현예들에 있어서, 각각의 호스트(104)에 대한 하드웨어 소켓 토폴로지들은 호스트(104) 내 각각의 리소스 그룹(200)의 각각의 메모리 액세스 토폴로지에 기초한다.

[0052]

[0057] NUMA 시스템에는, 프로세서들의 세트 및 메모리로 구성되는 다수의 NUMA 노드들이 있다. 앞서 개시된 바와 같이, 동일한 NUMA 노드(202-1) 내 프로세서(204)에 의한 메모리(206-1)로의 접속은 로컬적인 반면, 다른 NUMA 노드(202-2)의 메모리(206-2)에 접속하는 NUMA 노드(202-1)의 프로세서(204)는 원격이다. 일부 구현예들에 있어서, 원격 접속은 로컬 접속에 비해 다수의 사이클들을 취할 수 있는데, 이는 원격 접속이 다중-홉 동작을 수반할 수 있기 때문이다. 이러한 비대칭 메모리 접속 지연 때문에, 메모리 접속을 로컬로 유지하거나 메모리 로컬리티를 최대화하면 분산 처리에 있어서의 성능을 개선할 수 있다. 일부 구현예들에 있어서, NUMA 로컬리티를 활용함과 함께 NUMA 노드들 간에 CPU 로드 밸런싱을 수행하면 추가적인 성능 개선들로 이어질 수 있다.

[0053]

[0058] 마스터 제어기(108)는 작업 사양에 있어서의 하나 이상의 제한들을 인코딩하도록 구성된다(308). 예를 들어, 작업 사양(300)은 요청(112)의 매개변수들 및 값들로부터 유래하는 스케줄링 제한들을 포함할 수 있다. 예를 들어, 제어기(108)는 요청(112)의 매개변수들을, 특정 클라우드 영역 내에 위치되는 호스트 기계들에서의 작업에 대한 컴퓨테이션들을 수행하는 데 사용되는 데이터를 로드하도록 호스트(104)에 명령하는 작업 제한들로 변환할 수 있다. 예를 들어, 클라우드 영역은 워크로드의 주어진 작업에 대한 데이터 컴퓨테이션들을 수행하는데 필요한 소정 세트의 하드웨어 가속기 리소스들을 포함하는 데이터센터의 특정 물리적 또는 지리적 위치일 수 있다.

[0054]

[0059] 컴퓨팅 클러스터들(102)은 컴퓨팅 클러스터(102)의 다수의 호스트들(104) 간에 리소스 로컬리티를 활용하는 방식으로, 작업들을 스케줄링하고 호스트(104)의 특정 리소스 그룹들(200)에 배정하기 위한 배정 방식을 결정하도록 구성된다. 도 3에 도시된 예시의 작업 사양(300)은 16 개의 작업들을 실행하라는 요청을 수신하는 컴퓨팅 클러스터(102)를 나타내는 단순화된 작업 사양을 제공한다. 작업들 각각은 2 개의 NUMA 노드들을 포함하는 호스트(104)를 사용한다. 본 예에서, 작업들 각각은 96 개의 CPU들 및 4 개의 특수 목적 회로들, 예를 들어 하드웨어 가속기들의 전체 리소스 할당을 필요로 한다.

[0055]

[0060] 이러한 작업 사양(300)은 호스트(104)가 특정 NUMA 노드로부터 그 CPU 코어들을 할당하도록 하는 방법과 같이 특정 호스트(104)의 리소스 할당을 규정하는 매개변수들을 포함한다(310). 도 3의 예는 각각의 작업에 대해 96 개의 CPU들의 전체 리소스 할당을 만족하도록, 각각의 NUMA 노드로부터 할당 중인 48 개의 프로세서 코어들의 균형잡힌 CPU 할당을 보여준다. 다른 예들에 있어서, 제어기(108)는 제1 NUMA 노드로부터의 36 개의 CPU 들 및 제2 NUMA 노드로부터의 60 개의 CPU들과 같이 불균형 할당을 명시하는 작업 사양을 생성할 수 있다.

[0056]

[0061] 도 4는 기계-학습 워크로드들을 수행하도록 실행되는 작업들을 스케줄링하기 위한 예시의 공정(400)을 도시한다. 공정(400)은 전술한 시스템(100)을 사용하여 구현되거나 실행될 수 있다. 따라서, 공정(400)의 설명은 본 명세서에 설명된 다른 구성요소들 뿐만 아니라 시스템(100)의 전술한 컴퓨팅 리소스들을 참고할 수 있다. 일반적으로, 공정(400)의 설명들에 있어서 컴퓨팅 스텝들 또는 공정 흐름들은 다른 순서들로 발생하도록 그룹화되거나 배열될 수 있으며, 본 명세서에서 설명된 수치적인 차례로 제한되지 않는다.

[0057]

[0062] 이제 공정(400)을 참고하면, 시스템(100)은 그 컴퓨팅 클러스터들 중 하나 이상을 사용하여 워크로드를 수행하라는 요청을 수신한다(402). 일부 구현예들에 있어서, 공정(400)은 호스트(104)의 하드웨어 가속기들 및 다른 리소스들을 사용하여 워크로드를 수행하도록 작업들을 스케줄링하고 리소스들을 할당하기 위한 방법에 대응한다. 일부 예들에 있어서, 워크로드는 비디오 트랜스코딩, 이미지 처리, 음성 처리, 자율 주행 차량 내비게이션 또는 이미지 인식과 같은 특정 기계-학습 동작에 관련되는 트레이닝 또는 추론 워크로드이다.

[0058]

[0063] 요청(112)은 이미지에서 객체를 감지하거나, 음성 발언에서 용어들을 인식하기 위해 추론 워크로드와 같은 ML 워크로드를 수행하는 것일 수 있다. 이러한 맥락에서, 하드웨어 가속기들 중 하나 이상은 컨볼루션 신경망(CNN) 또는 순환 신경망(RNN)과 같은, 다수의 신경망 레이어들을 포함하는 신경망을 구현하도록 구성될 수 있다. 수신된 요청(112)은 워크로드의 작업들을 실행하는 데 사용되어야 하는 특정 유형의 신경망 구성(예를 들

어, CNN 또는 RNN)을 명시하는 매개변수들을 포함할 수 있다.

[0059] [0064] 수신된 요청(112)에는, 예를 들어 호스트(104)의 리소스 그룹을 사용하여 클러스터(102)에 특정 신경망을 배치하기 위해 제2 요청(112)이 뒤따를 수도 있다. 제2 요청에는 제어기(108)(또는 호스트(104))가 특정 신경망 레이어에 대한 가중치들의 세트에 대한 매개변수들을 취득하도록 하는 명령들 또는 지시들이 뒤따를 수 있다. 예를 들어, 가중치들의 세트는 명령들에 명시된 위치 주소들에 기초하여, 호스트(104)에 의해 관리되는 메모리의 위치들로부터 취득될 수 있다. 일부 구현예들에 있어서, 호스트(104)에 의해 취득되는 가중치들을 저장하는 메모리는 호스트(104)에서 리소스 그룹을 규정하는 NUMA 노드의 다수의 로컬 리소스들 중 하나이다. 유사하게, 명령들은 제어기(108)(또는 호스트(104))가 다른 메모리 위치들에 접속하여, 신경망 레이어를 통해 처리하기 위한 입력들을 가져와 호스트(104) 내 NUMA 노드의 로컬 리소스들(105)을 사용하여 신경망 레이어에 대한 출력을 생성하도록 할 수 있다. 일부 구현예들에 있어서, 요청(112)에서 식별된 입력들의 특정 부분을 신경망 레이어를 통해 처리하여 레이어 출력을 생성하는 것은 컴퓨팅 클러스터(102)의 다수의 호스트들(104)에 걸쳐 또는 다수의 컴퓨팅 클러스터들(102)에 걸쳐 처리될 수 있는 더 큰 워크로드의 하나 이상의 작업들의 실행을 나타낼 수 있다.

[0060] [0065] 시스템(100)은 요청에 기초하여 리소스 요건을 결정한다(404). 리소스 요건은 ML 워크로드를 나타내는 작업들의 세트를 수행하는 데 필요한 컴퓨터이셔널 리소스들의 유형들 및 수량과 같은, 워크로드 요청(112)에 대한 시스템(100)의 리소스들에 대한 소정의 상세들을 표시할 수 있다. 예를 들어, 리소스 요건은 분산 시스템에 있어서, 소정의 프로세서 또는 프로세서 유형, 처리 능력 또는 속도, 메모리의 양 또는 메모리 크기, 하드웨어 가속기들의 수량 또는 리소스 로컬리티의 측정을 명시할 수 있다.

[0061] [0066] 시스템(100)은 각각의 호스트에 대한 다수의 하드웨어 가속기들 및 리소스 요건에 기초하여, ML 워크로드의 각각의 작업을 실행하도록 배정되는 호스트들(104)의 수량을 결정한다(406). 그 수량의 호스트들(104) 각각의 호스트(104)에 대해, 시스템(100)은 호스트의 메모리 액세스 토폴로지에 기초하여 각각의 작업 사양을 생성한다(408). 호스트의 메모리 액세스 토폴로지는 호스트(104)의 각각의 리소스 그룹(200)에 대한 다수의 각각의 특정 NUMA 토폴로지들 중 하나에 기초할 수 있다. 일부 구현예들에 있어서, 리소스 그룹(200)의 특정 NUMA 토폴로지는 로컬 NUMA 노드에 특정되며, 그룹의 다른 리소스들에 대해 로컬적인 각각의 메모리(M)를 포함한다. 각각의 메모리(M)는 NUMA 노드의 하나 이상의 다른 리소스들 및 적어도 하나의 하드웨어 가속기에 로컬적으로 메모리를 결합하는 소켓 인터페이스를 포함할 수 있다.

[0062] [0067] 제어기(108)는, 요청(112)의 매개변수들을 스캐닝하고 그리고 제어기(108)에 슬레이브 자산들로서 배정된 호스트들(104)의 세트에서 각각의 리소스 그룹(200)에 대한 각각의 하드웨어 토폴로지를 상호 참조하는 것에 대한 응답으로, 작업 사양을 생성한다. 시스템(100)은 호스트에 각각의 작업 사양을 제공한다(410). 예를 들어, 제어기(108)는 다수의 각각의 작업 사양들을 다른 호스트들(104)에 제공할 수 있고, 시스템(100)은 호스트들 각각에 대한 각각의 작업 사양에서 명시되는 작업들을 실행함으로써 ML 워크로드를 수행한다(412).

[0063] [0068] 도 5는 시스템(100)의 호스트에 제공되는 작업 사양을 생성하기 위한 예시의 공정을 도시한다. 공정(400)과 마찬가지로, 공정(500)은 시스템(100)을 사용하여 구현되거나 실행될 수 있으며, 공정(500)에 대한 설명들은 본 명세서에서 설명되는 다른 구성요소들을 포함하여, 시스템(100)의 리소스들을 참조할 수 있다. 일반적으로, 공정(500)의 설명들에 있어서의 컴퓨팅 스텝들 또는 공정 흐름들은 다른 순서들로 발생하도록 그룹화되거나 배열될 수 있으며, 본 명세서에서 설명된 수치적인 차례로 제한되지 않는다.

[0064] [0069] 이제, 공정(500)을 참고하면, 시스템(100)은 각각의 호스트에 대한 NUMA 소켓들의 맵핑을 설명하는 시스템 파일을 사용하여 호스트의 리소스들에 대한 하나 이상의 소켓들을 식별(502)하도록 구성된다. 예를 들어, 제어기(108)는 호스트(104)에서 리소스 그룹들의 하드웨어 토폴로지에 기초하여, NUMA 소켓들의 맵핑을 결정하도록 구성된다. NUMA 소켓들의 맵핑은 호스트(104)에 의해 관리되는 리소스 그룹들 각각의 NUMA 토폴로지를 표시하도록 사용된다.

[0065] [0070] 제어기(108)는 호스트에 대한 NUMA 소켓들의 맵핑에서 설명되는 리소스들에 대한 소켓들을 사용하여 호스트(104)의 제어 그룹을 구성한다(504). 일부 구현예들에 있어서, 제어기(108)는 하나 이상의 프로토콜 비트들을 호스트(104)에 전달하고, 호스트의 스케줄러는 호스트(104)에서 다양한 리소스 그룹들 중 리소스들(105)의 로컬리티에 기초하여, 제어 그룹을 구성하기 위해 (예를 들어, 요청 또는 작업 사양의) 프로토콜 비트들을 사용한다. 예를 들어, 호스트(104)의 스케줄러 또는 제어기(108)는 수신된 요청(112)에서의 제한들의 일부(또는 전부)를 만족하는, 호스트(104)에 있어서의 리소스들의 유형들 및 리소스 그룹들의 각각의 NUMA 토폴로지에 기초하여 제어 그룹을 구성하도록 작동 가능하다.

- [0066] [0071] 제어기(108)는 작업 사양의 ML 작업들을 호스트(104)에서 구성되는 제어 그룹에 결합 또는 배정하기 위해 호스트(104)와 협동한다. 일부 구현예들에 있어서, 제어기(108)에 의해 호스트(104)로 전달되는 프로토콜 비트들은 작업 사양의 ML 작업들을 호스트(104)에서 구성되는 제어 그룹에 결합 또는 배정하기 위해 호스트(104)의 스케줄러에 의해 사용된다. 프로토콜 비트들은 작업 사양의 하나 이상의 제한들 또는 요건들을 표시하기 위해 작업 사양에 포함될 수 있다. 프로토콜 비트들은 하나 이상의 코드화된 신호들, 이진수 값들의 데이터 워드 또는 다른 관련 매개변수들 또는 데이터 값들에 의해 표현될 수 있다. 코드화된 신호들 및 이진수 값들은 작업 사양의 작업들의 배정을 결정하기 위해 호스트의 스케줄러에 의해 수신되어 처리될 수 있거나, 달리 해석될 수 있다. 일부 구현예들에 있어서, 프로토콜 비트들은 작업 사양과 관련될 수 있으나, 작업 사양과는 분리되어 제공될 수 있다.
- [0067] [0072] 호스트(104)는 제어 그룹의 리소스들의 사용 가능성을 포함하여 제어 그룹의 특정 유형의 가속기들에 대한 소켓 인터페이스들에 적어도 기초하여, 작업 사양의 ML 작업들을 제어 그룹에 결합(506)하도록 작동 가능하다. 호스트(104)는 제어 그룹 하의 공정으로서 작업 사양의 ML 작업들을 실행하기 위해 제어 그룹의 메모리 리소스들 및 하드웨어 가속기들을 사용한다(508). 일부 구현예들에 있어서, 호스트(104)는 로컬 메모리를 포함하여, ML 작업들을 실행하기 위해 제어 그룹의 하나 이상의 다른 리소스들을 사용할 수 있다. 일부 예들에서, 비-로컬 메모리가 활용될 수 있으나, 이러한 리소스들을 활용할 때 필요한 컴퓨팅 대역폭 또는 발생할 수 있는 임의의 성능 영향들과 균형을 맞출 수 있다.
- [0068] [0073] 본 명세서에서 설명된 청구 대상 및 기능적 동작들의 실시예들은 본 명세서에 개시된 구조들 및 이들의 구조적 균등물들을 포함하여, 디지털 전자 회로에서, 유형적으로(tangibly) 구현되는 컴퓨터 소프트웨어 또는 펌웨어에서, 컴퓨터 하드웨어에서, 또는 이를 중 하나 이상의 조합들로 구현될 수 있다. 본 명세서에서 설명된 청구 대상의 실시예들은 하나 이상의 컴퓨터 프로그램들, 즉 데이터 처리 장치에 의한 실행을 위해, 또는 데이터 처리 장치의 동작을 제어하기 위해, 유형의 비 일시적 프로그램 캐리어 상에서 인코딩되는 컴퓨터 프로그램 명령들의 하나 이상의 모듈들로서 구현될 수 있다.
- [0069] [0074] 대안적으로 또는 추가적으로, 프로그램 명령들은 인공적으로 생성된 전파 신호, 예를 들어 기계로 생성된 전기, 광학 또는 전자기 신호에 인코딩될 수 있으며, 이는 데이터 처리 장치에 의한 실행을 위해 적합한 수신기 장치로 전송하기 위한 정보를 인코딩하기 위해 생성된다. 컴퓨터 저장 매체는 기계-판독 가능한 저장 디바이스, 기계-판독 가능한 저장 기판, 랜덤 또는 시리얼 액세스 메모리 디바이스 또는 이를 중 하나 이상의 조합일 수 있다.
- [0070] [0075] 본 명세서에서 설명된 공정들 및 로직 흐름들은, 하나 이상의 프로그래밍 가능한 컴퓨터들이 입력 데이터에 대해 동작하여 출력(들)을 생성함으로써 기능들을 수행하기 위해 하나 이상의 컴퓨터 프로그램들을 실행하여 수행할 수 있다. 공정들 및 로직 흐름들은 또한 특수 목적 로직 회로, 예를 들어 FPGA(field programmable gate array), ASIC(application specific integrated circuit) 또는 GPGPU(General purpose graphics processing unit)에 의해 수행될 수 있으며, 장치는 또한 이를로서 구현될 수 있다.
- [0071] [0076] 컴퓨터 프로그램의 실행에 적합한 컴퓨터들은 예로서, 범용 또는 특수 목적 마이크로프로세서들 또는 둘 모두, 또는 임의의 다른 종류의 중앙 처리 장치를 포함하고, 이에 기초할 수 있다. 일반적으로, 중앙 처리 장치는 판독 전용 메모리 또는 랜덤 액세스 메모리 또는 둘 모두로부터 명령들 및 데이터를 수신할 것이다. 컴퓨터의 필수 요소들은 명령들을 수행하거나 실행하기 위한 중앙 처리 장치 및 명령들 및 데이터를 저장하기 위한 하나 이상의 메모리 디바이스들이다. 일반적으로, 컴퓨터는 또한, 데이터를 저장하기 위한 하나 이상의 대용량 저장 디바이스들, 예를 들어 자기, 자기 광 디스크들 또는 광 디스크들을 포함하거나, 이들로부터 데이터를 수신하거나 이들로 데이터를 전송하거나 또는 이 둘 모두를 하도록 작동 가능하게 결합될 것이다. 그러나, 컴퓨터는 이러한 디바이스들을 가질 필요는 없다.
- [0072] [0077] 컴퓨터 프로그램 명령들 및 데이터를 저장하기에 적합한 컴퓨터 판독 가능한 매체는, 예로서 반도체 메모리 디바이스들, 예를 들어 EEPROM, EEPROM 및 플래시 메모리 디바이스들; 자기 디스크들, 예를 들어 내부 하드 디스크들 또는 분리성 디스크들을 포함하여 모든 형태들의 비휘발성 메모리, 매체 및 메모리 디바이스들을 포함한다. 프로세서 및 메모리는 특수 목적 로직 회로에 의해 보충되거나 이에 통합될 수 있다.
- [0073] [0078] 본 명세서가 많은 구체적인 구현예의 상세들을 포함하고 있지만, 이들은 모든 발명 또는 청구될 수 있는 것의 범주에 대한 제한들로 해석되어서는 아니되며, 오히려 특정 발명들의 특정 실시예들로 특정될 수 있는 특징들에 대한 설명들로 해석되어야 한다. 별도의 실시예들의 맥락에 있어서 본 명세서에서 설명되는 소정의 특징들은 단일 실시예에서의 조합으로 구현될 수도 있다. 반대로, 단일 실시예의 맥락에서 설명되는 다양한 특징

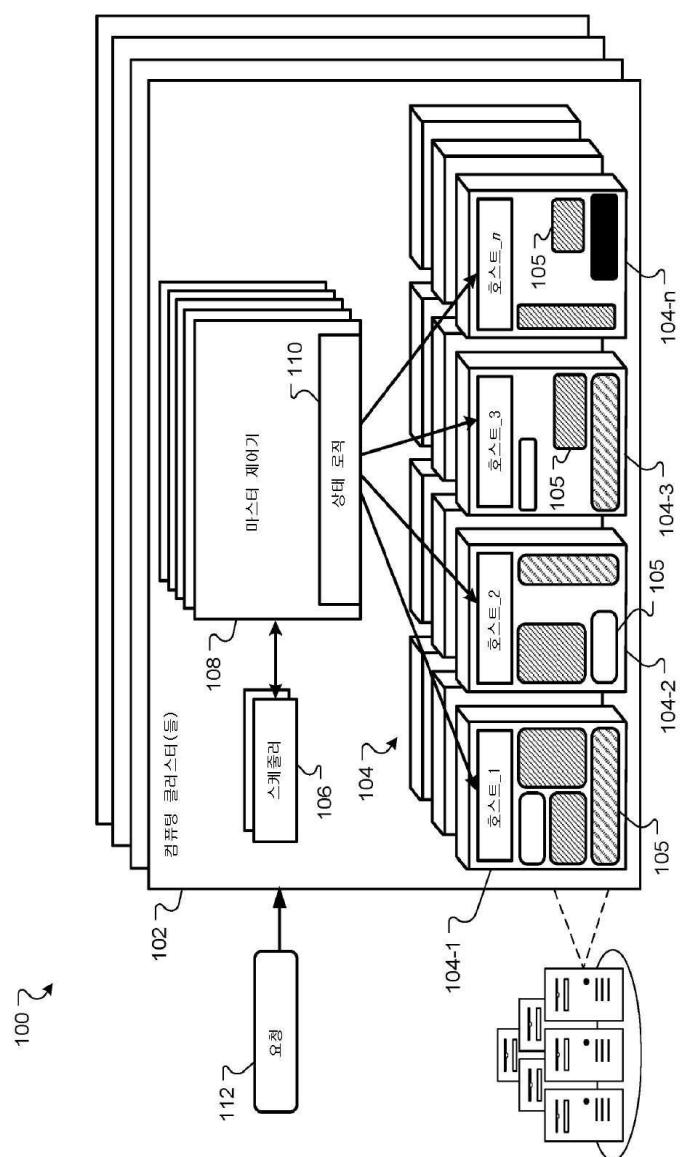
들은 별도로 또는 임의의 적합한 하위 조합으로 다수의 실시예들에서 또한 구현될 수 있다. 또한, 비록 특징들이 소정의 조합들로 작용하는 것으로 전술될 수 있고, 심지어 초기에 그렇게 청구될 수 있지만, 일부 케이스들에 있어서는 청구된 조합으로부터의 하나 이상의 특징들이 그 조합으로부터 제거될 수 있으며, 청구된 조합은 하위 조합 또는 하위 조합의 변형으로 유도될 수 있다.

[0074] 유사하게, 도면들에서는 동작들이 특정 순서로 도시되어 있지만, 이는 바람직한 결과들을 달성하기 위해 그러한 동작들이 도시된 특정 순서로 또는 순차적인 순서로 수행되거나, 예시된 모든 동작들이 수행되어야 함을 요구하는 것으로 이해되어서는 아니된다. 소정의 상황들에서는, 멀티태스킹 및 병렬 처리가 유리할 수 있다. 또한, 전술된 실시예들에서 다양한 시스템 모듈들 및 구성요소들의 분리는 모든 실시예들에서 그러한 분리를 요구하는 것으로 이해되어서는 아니되며, 설명된 프로그램 구성요소들 및 시스템들은 일반적으로 단일 소프트웨어 제품으로 함께 통합되거나, 다수의 소프트웨어 제품들로 패키징될 수 있다는 것이 이해되어야 한다.

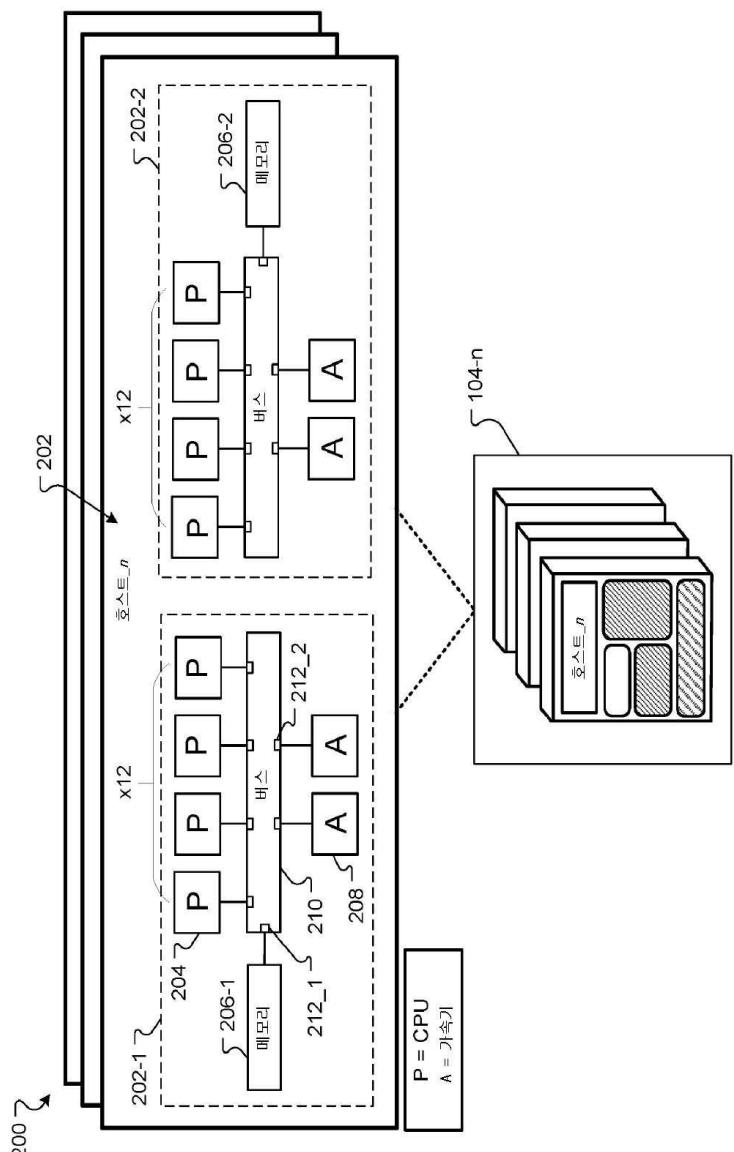
[0075] 청구 대상의 특정 실시예들이 설명되었다. 다른 실시예들은 이하의 청구범위의 범주 내에 있다. 예를 들어, 청구범위에서 언급된 조치들은 다른 순서로 수행될 수 있으며, 여전히 바람직한 결과들을 얻을 수 있다. 일례로서, 첨부 도면들에 도시된 공정들은 바람직한 결과를 달성하기 위해, 도시된 특정 순서 또는 순차적인 순서를 반드시 필요로 하지는 않는다. 소정의 구현예들에 있어서, 멀티태스킹 및 병렬 처리가 유리할 수 있다.

도면

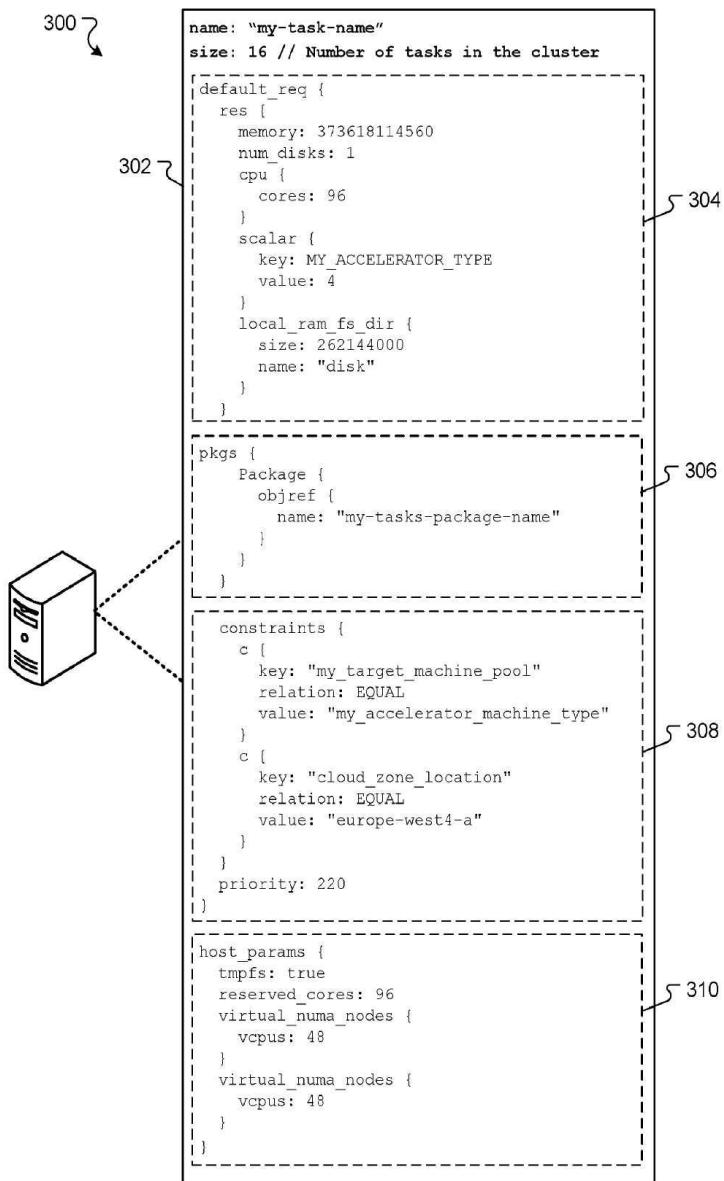
도면1



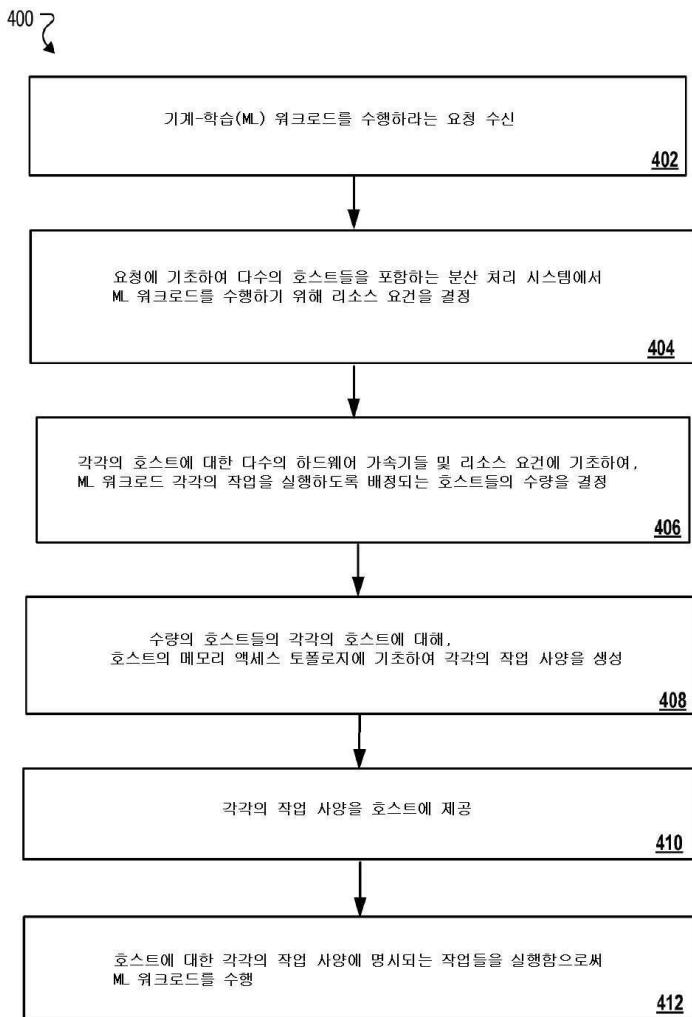
도면2



도면3



도면4



도면5

500
↳