대규모 딥러닝 고속처리를 위한 분산 딥러닝 플랫폼

목 차

1. 기술의 개요

▣ 분산 딥러닝 플랫폼

❖ 기술 정의 : HPC(고성능컴퓨팅) 시스템 상에서 다수의 서버들을 이용하여 더 빠르게, 더 효율적으로 대규모 딥러닝 모델의 학습을 수행하는 S/W

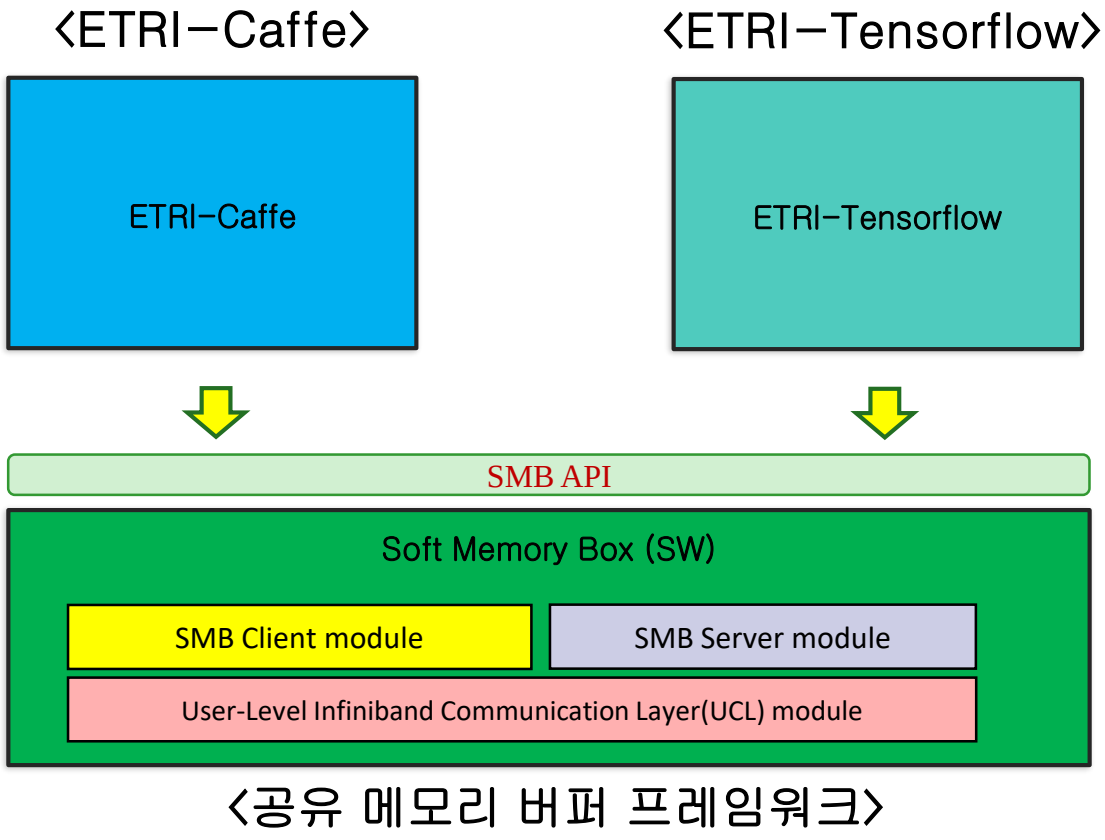
❖ 기술 분야 : 딥러닝, 분산처리, 병렬처리

❖ 기술 개발 배경

- 딥러닝 기술은 높은 정확도를 요구하는 딥러닝 모델일수록 더 많은 학습데이터와 더 높은 해상도의 학습 데이터를 요구(예, 고해상도 영상 처리 요구 증가)
- 더 높은 정확도를 가지는 모델은 기하급수적인 계산량 증가를 수반하며, HPC 시스템을 이용하여 대규모 딥러닝 모델을 분산 학습하려는 수요 증가
- 다수의 서버를 이용한 딥러닝 분산 학습은 대규모 통신이 필요하여 통신 병목이 발생함. 이를 해결하는 고속 분산 병렬 학습 기술이 필요
- 관련 용어

– HPC : High Performance Computing

▣ 기술이전 내용 및 범위



<공유 메모리 버퍼 프레임워크>

2. 기술이전 내용 및 범위

▣ 기술이전 범위

❖ [SW] 분산 딥러닝 프로그램 3종

1) 통합 공유 메모리 버퍼 프레임워크 SW 버전 1.0
2) 공유메모리 기반 분산 딥러닝 지원 에트리(ETRI) 카페(Caffe) 버전 3.0
3) 공유 메모리 기반 에트리-텐서플로우(ETRI-TensorFlow) 2.5

❖ [문서] 대시보드 설계문서 8종

1) 분산 딥러닝 플랫폼 요구사항정의서
2) 딥러닝 HPC 분산 딥러닝 플랫폼 상세설계서
3) 딥러닝 HPC 분산 딥러닝 플랫폼 시험절차서
4) 딥러닝 HPC 분산 딥러닝 플랫폼 시험결과서
5) 소프트 메모리 박스 사용자 매뉴얼
6) ShmCaffe 분산처리 확장성 분석
7) ETRI-Caffe 사용자 매뉴얼
8) ETRI-Tensorflow 사용자 매뉴얼

❖개술 개발 현황

기술성숙도(TRL : Technology Readiness Level) **단계 : (6)단계**

2. 기술이전 내용 및 범위

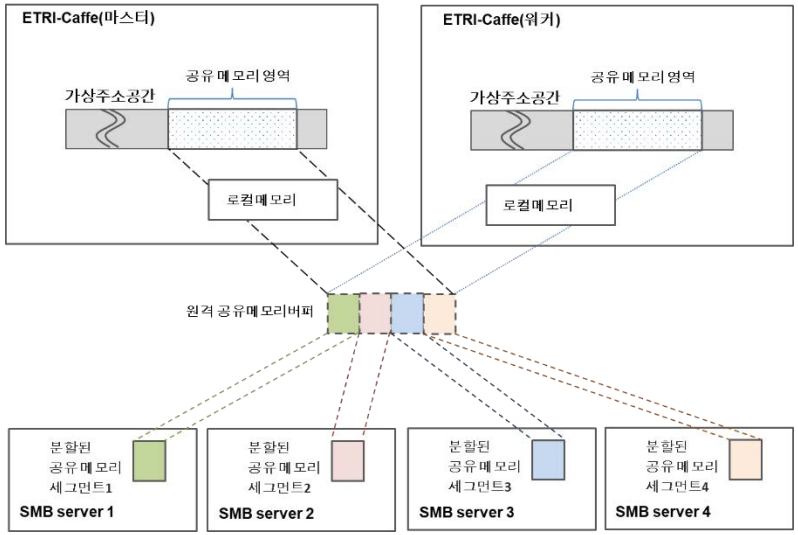
▣ 소프트 메모리 박스 기술 현황

❖ 통합 공유 메모리 버퍼 프레임워크

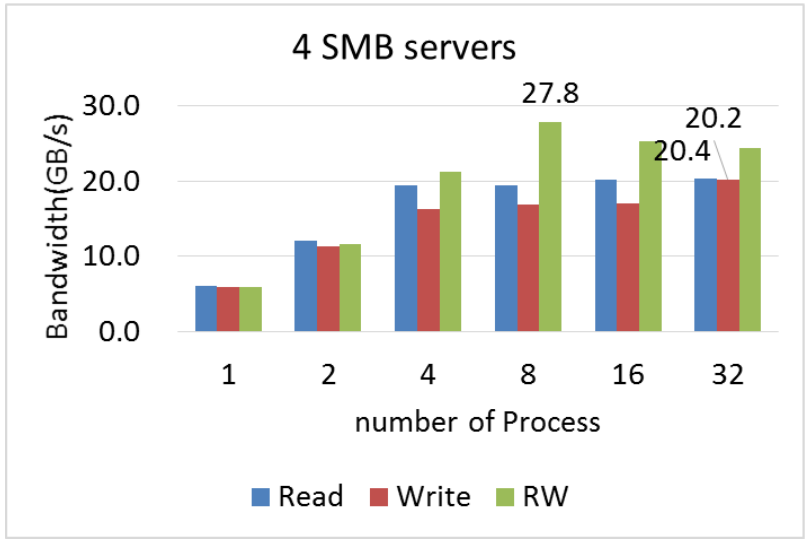
- 다중 서버 공유 메모리 통합 기술
- 원격 메모리 직접 읽기/쓰기
- 분산 공유메모리 할당/해제/접근/잠금
- 정적/공유 라이브러리 제공

❖ 고속 딥러닝 파라미터 통신 제공

- Throughput: 7GB/1SMB server
  → 27.8GB/4SMB servers
- 우수한 확장 효율: 99%/4SMB servers



<통합 분산 공유 메모리 프레임워크 구조>



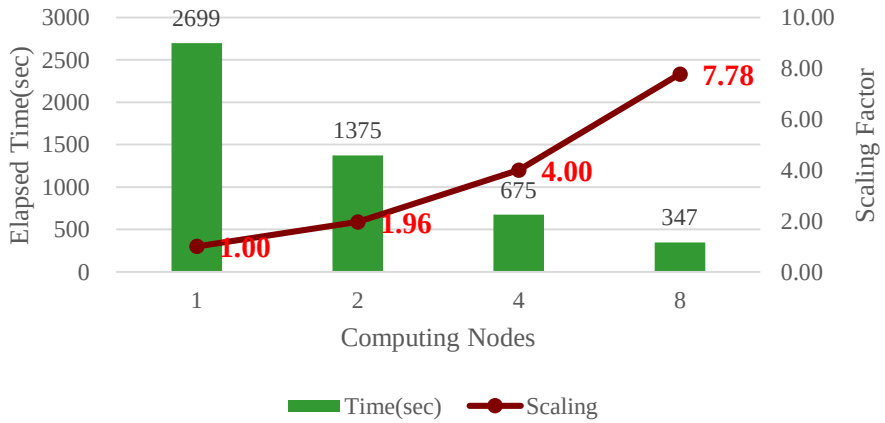
< SMB R/W Throughput>

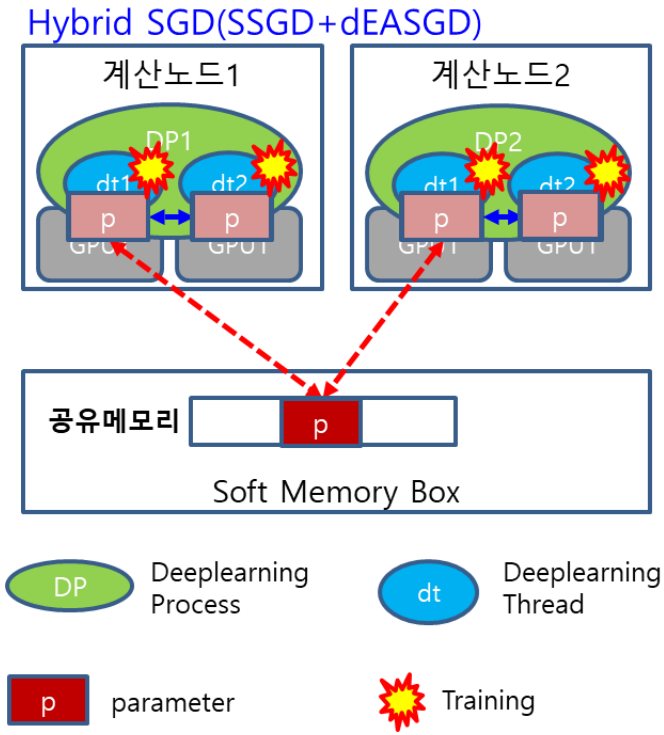
2. 기술이전 내용 및 범위

▣ ETRI-Caffe 기술 현황

- BVLC/Nvidia Caffe 모델 호환
- MPI 기반 분산 딥러닝 실행 관리
- 비동기식/하이브리드 파라미터 업데이트
- 고속 딥러닝 분산 트레이닝
- 우수한 노드 확장성(97% 확장 효율, 1→8 node)

<이미지 인식 분산 처리 확장성>





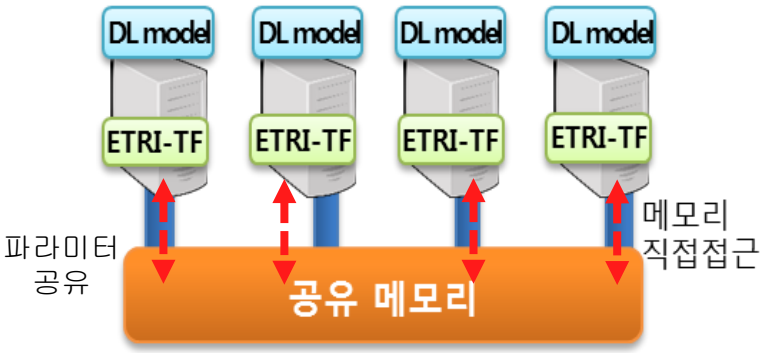
<하이브리드 분산 병렬 학습 방식>

BVLC : Berkeley Vision and Learning Center, MPI: Message Passing Interface

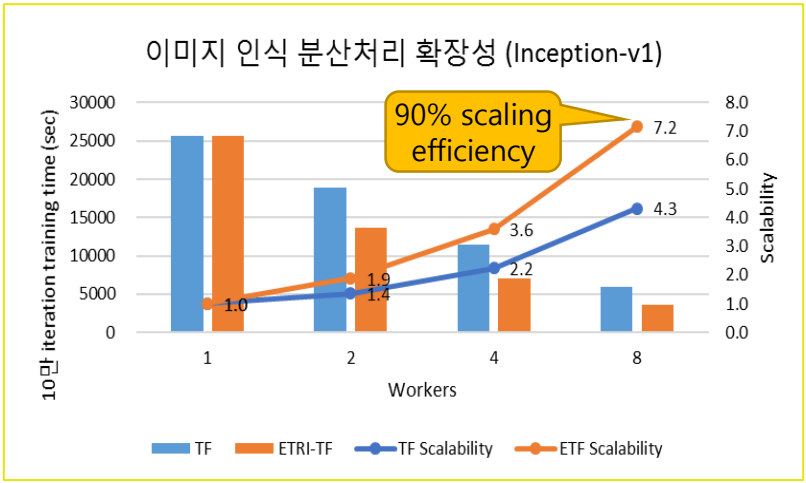
2. 기술이전 내용 및 범위

▣ ETRI-Tensorflow 기술 개발 현황

- TensorFlow 모델 호환
- CNN, RNN 및 그 외 DNN 모델 지원
- 데이터 병렬 및 모델 병렬 분산 트레이닝 지원
- 비동기식 데이터/모델 병렬 트레이닝
- 공유메모리 기반 분산 트레이닝 가속
- TensorFlow 대비 2배 빠른 학습
- 다수 이미지 인식, 음성인식 모델로 성능 검증
- 우수한 확장 성능 제공 (90% 확장 효율)



<ETRI-TensorFlow 기술 개념도>

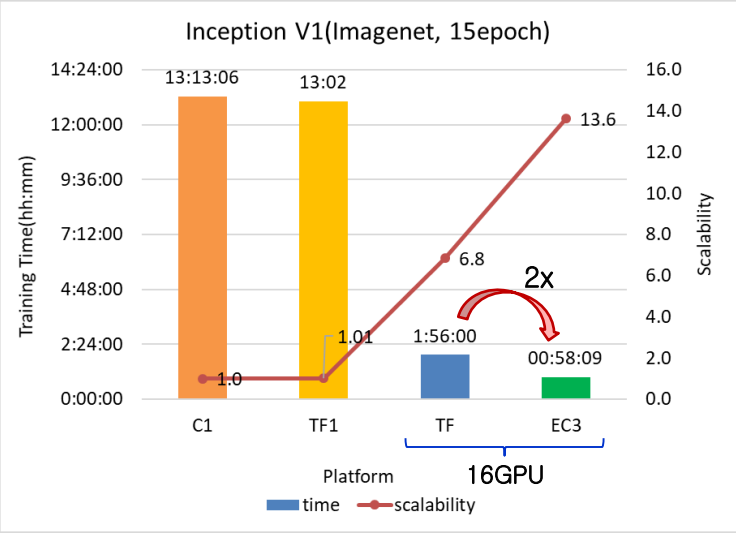


3. 경쟁기술과 비교

▣ 경쟁 분산 딥러닝 기술과의 성능 비교

❖ ETRI-Caffe

- NVIDIA Caffe(1GPU)대비 13.6배(16GPU)
- TF 대비 최대 2배 빠른 학습



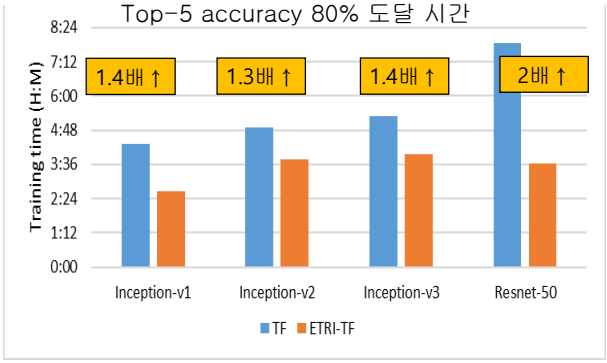
C1: Caffe(1GPU), TF1: Tensorflow(1GPU)
TF: Tensorflow(v1.13) EC3: ETRI-Caffe(v3.0)

< ETRI-Caffe 성능 비교 >

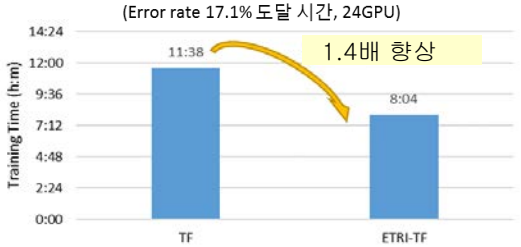
❖ ETRI-Tensorflow

- 이미지 인식 : Tensorflow 대비 최대 2배
- 음성인식 트레이닝: 최대 1.4배

<이미지 인식 모델 트레이닝 성능>



<음성인식 모델 트레이닝 성능>



< ETRI-Tensorflow 성능 비교>

4. 기술의 사업성

▣ 사업 가능 영역

- ❖ (B2B) 기업용 온프레미스 분산 딥러닝 개발 환경/인프라 구축
- ❖ (B2B/B2C) 클라우드기반 분산 딥러닝 개발 환경 서비스

(수요처)
병원, 중견기업,
공공기관,
대학교 연구실,

분산 딥러닝 플랫폼 기술

(B2B) 기업용 온프레미스
분산 딥러닝 개발환경 구축

(수요처)
클라우드
서비스 기업

분산 딥러닝 플랫폼 기술

(B2B/B2C) 클라우드기반
분산 딥러닝 개발 환경

4. 기술의 사업성

▣ 기술이전 방식

❖ 정액기술료(세부기술별 이전 가능)

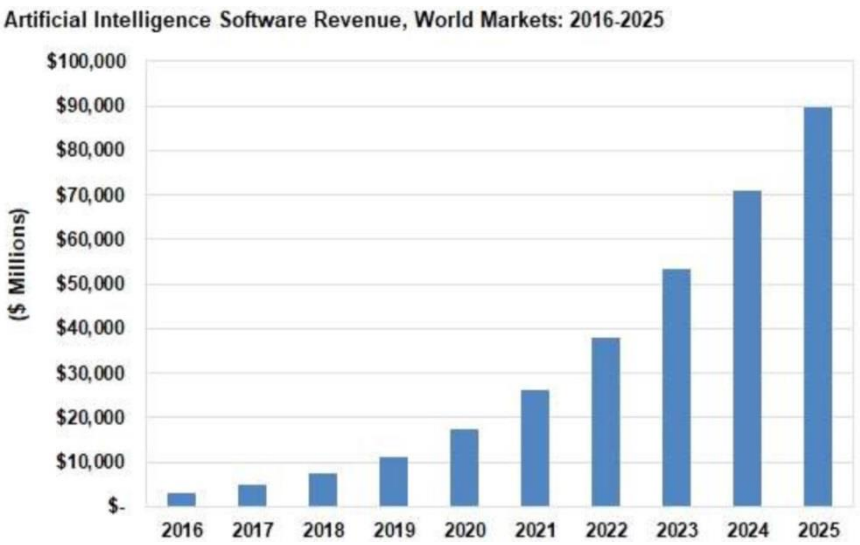
구분		착수기본료(원)		
		중소기업	중견기업	대기업
A. Soft Memory Box	특허2건실시권 소스프로그램1건 기술문서5건	50,000,000	150,000,000	200,000,000
B. ETRI-Caffe	특허1건실시권 소스프로그램1건 기술문서2건	13,000,000	39,000,000	52,000,000
C. ETRI-Tensorflow	특허1건실시권 소스프로그램1건 기술문서1건	20,000,000	60,000,000	80,000,000
합계	특허4건, 프로그램3건, 기술문서 8건	83,000,000	249,000,000	332,000,000

- A기술은 B,C 기술을 활용하기 위한 필수 기술임

5. 국내외 시장 동향

▣ 국외 AI HW/SW 시장 예측 (2017-2025)

글로벌 AI시장 규모 및 전망



Source: Tractica, 2017

인공지능 시장 전망

- 2017년 구글 연례 개발자 회의에서 구글 CEO 선다 피차이는 '약 10년 주기로 PC에서 웹, 스마트폰으로 컴퓨터 주류 형태가 변해왔는데, 이제 모바일 퍼스트에서 AI퍼스트로 옮겨가고 있다'고 밝힘. 엄청난 속도로 쌓이는 데이터를 가치 있는 비즈니스로 만들어주는 도구가 AI임
- 전 세계 인공지능 기반 스마트 머신 시장은 2014년 62억 2,900만 달러에서 2019년 152억 7,900만 달러 규모로 성장 전망됨 (BCC리서치)
- 영상처리 시장은 2015년 765억 달러에서 2017년 기준 1,090억 달러, 음성인식 시장은 같은 기간 840억 달러에서 1,130억 달러 수준으로 성장 예상됨

5. 국내외 시장 동향

▣ 국내 AI 산업 시장 예측 (2014-2020, 단위 조원)

AI 분야별 국내 시장규모 및 전망

Source : 과학기술정보통신부

영상 분석 산업 전망

- 2020년 영상처리 시스템 세계 시장규모는 약 176억 달러로 전망되며, 2013년~2015년 연 성장률 8.80%를 보였음 (ETRI 기술경제연구본부, 2016)
- 국내 영상처리 시스템 시장 규모 역시 2013~2015년 연 4% 이상의 성장률로 2020년 1,900억 원을 돌파할 것으로 예상됨
- 카메라 기술의 빠른 발전으로 영상 및 이미지 해상도가 높아지는 가운데, 이를 빠르게 분석, 처리할 수 있는 GPU의 등장으로 영상 및 이미지 분석 시장 성장이 가속될 것으로 보임

5. 국내외 시장 동향

▣ 예상 제품/서비스의 예상매출액(생산/판매부터 향후 매 5년 간 추정)

(단위: M$(국외), 십억원(국내))

관련제품/서비스	시장	2020	2021	2022	2023	2024	합계
기업용 딥러닝 개발 환경 (on-premise용 플랫폼)	국외	0.00	132.00	420.00	846.00	1551.00	2949.00
	국내	12.24	6.66	10.99	20.15	35.46	85.50
클라우드 기반 딥러닝 개발 환경 (클라우드용 플랫폼)	국외	0.00	44.00	140.00	282.00	413.60	879.60
	국내	0.41	1.06	2.52	5.64	9.93	19.55
합계	국외	0.00	176.00	560.00	1128.00	1964.60	3828.60
	국내	12.65	7.72	13.51	25.79	45.39	105.05


감사합니다.

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년08월25일
(11) 등록번호 10-2292389
(24) 등록일자 2021년08월17일

(51) 국제특허분류(Int. Cl.)
G06F 12/084 (2016.01) *G06F 12/02* (2018.01)
G06F 13/28 (2006.01)
(52) CPC특허분류
G06F 12/084 (2013.01)
G06F 12/0292 (2013.01)
(21) 출원번호 10-2018-0006024
(22) 출원일자 2018년01월17일
심사청구일자 2019년03월14일
(65) 공개번호 10-2019-0087783
(43) 공개일자 2019년07월25일
(56) 선행기술조사문헌
KR1020160033505 A*
KR1020130079865 A*
JP2013513839 A*
KR101533405 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
(72) 발명자
안신영
대전광역시 서구 둔산북로 160, 5동 701호
임은지
대전광역시 유성구 노은동로 187, 602동 1801호
(뒷면에 계속)
(74) 대리인
한양특허법인

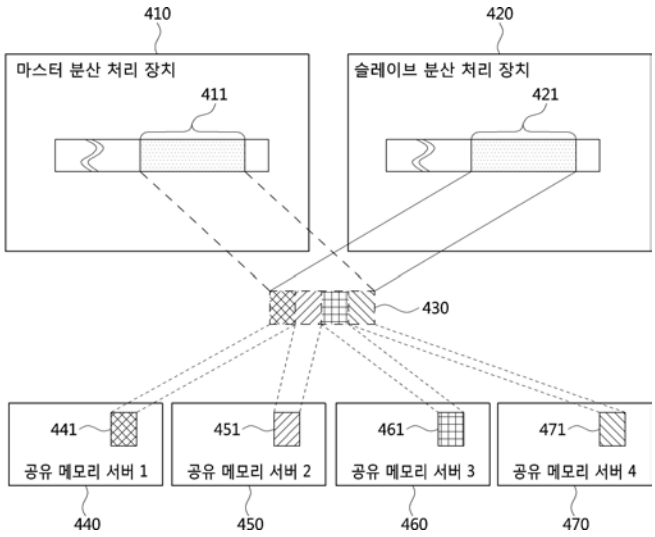
전체 청구항 수 : 총 19 항 심사관 : 김계준

(54) 발명의 명칭 **원격 직접 메모리 접근을 통한 분산 처리 장치 및 그 방법**

(57) 요 약

본 발명의 일 실시예는, 공유 메모리 서버 클러스터를 구성하는 복수의 공유 메모리 서버들에 구비된 원격 메모리들에 직접 접근하여 데이터를 송수신하는 통신부; 상기 원격 메모리들로부터 할당된 공유 메모리 버퍼 세그먼트들로 구성된 공유 메모리 버퍼에 대하여, 메모리에 상기 공유 메모리 버퍼와 동일한 크기만큼 로컬 공유 메모리 영역을 할당하고, 상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역을 동기화하는 공유 메모리 접근 관리부; 상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역 사이의 메모리 맵핑 테이블을 관리하는 메모리 맵핑 테이블 관리부; 및 상기 로컬 공유 메모리 영역에 대한 주어진 연산을 수행하는 연산부를 포함하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치를 제공한다.

대 표 도



(52) CPC특허분류
G06F 13/28 (2013.01)
(72) 발명자
최용석
전광역시 유성구 지족북로 60, 207동 303호
우영춘
대전광역시 유성구 어은로 57, 113동 404호
최완
대전광역시 서구 관저북로 52, 108동 306호

이 발명을 지원한 국가연구개발사업

과제고유번호	2016-0-00087
부처명	미래창조과학부
과제관리(전문)기관명	정보통신기술진흥센터(IITP)
연구사업명	정보통신방송기술개발사업(SW컴퓨팅산업원천기술개발사업)
연구과제명	대규모 딥러닝 고속 처리를 위한 HPC 시스템 개발
기 여 율	1/1
과제수행기관명	한국전자통신연구원
연구기간	2017.01.01 ~ 2017.12.31

명 세 서

청구범위

청구항 1

공유 메모리 서버 클러스터를 구성하는 복수의 공유 메모리 서버들에 구비된 원격 메모리들에 직접 접근하여 데이터를 송수신하는 통신부;

상기 원격 메모리들로부터 할당된 공유 메모리 버퍼 세그먼트들로 구성된 공유 메모리 버퍼에 대하여, 로컬 메모리에 상기 공유 메모리 버퍼와 동일한 크기만큼 로컬 공유 메모리 영역을 할당하고, 상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역을 동기화하는 공유 메모리 접근 관리부;

상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역 사이의 메모리 맵핑 테이블을 관리하는 메모리 맵핑 테이블 관리부; 및

상기 로컬 공유 메모리 영역에 대한 주어진 연산을 수행하는 연산부를 포함하고,

상기 공유 메모리 버퍼 세그먼트들은 서로 다른 상기 공유 메모리 서버들에 의해 할당되고,

상기 공유 메모리 버퍼는 복수의 분산 처리 장치들에 의하여 공유되는 가상의 연속된 버퍼에 상응하고,

상기 로컬 공유 메모리 영역은 가상주소만을 반환하는 것이 아니고 실제 물리 메모리에 할당되는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치.

청구항 2

청구항 1에 있어서,

상기 공유 메모리 접근 관리부는

직접 상기 공유 메모리 버퍼를 생성하여 동일한 분산 처리 프레임워크를 구성하는 다른 분산 처리 장치들에 공유 메모리 버퍼 정보를 공유하거나, 상기 다른 분산 처리 장치에 의하여 생성된 상기 공유 메모리 버퍼에 상응하는 공유 메모리 버퍼 정보를 수신하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치.

청구항 3

청구항 2에 있어서,

상기 공유 메모리 접근 관리부는

상기 공유 메모리 서버들에 각각에 상응하는 상기 공유 메모리 버퍼 세그먼트들의 크기를 계산하고, 상기 공유 메모리 서버들에 상기 공유 메모리 버퍼 세그먼트들의 생성 및 할당을 요청하여 상기 공유 메모리 버퍼를 생성하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치.

청구항 4

청구항 2에 있어서,

상기 공유 메모리 접근 관리부는

상기 연산에 의하여 상기 로컬 공유 메모리 영역의 데이터가 변경된 경우에 상기 로컬 공유 메모리 영역의 데이터를 상기 공유 메모리 버퍼에 복사하여 상기 원격 메모리들과 데이터를 동기화하고, 상기 다른 분산 처리 장치들에 의하여 상기 공유 메모리 버퍼의 데이터가 변경된 경우에 상기 공유 메모리 버퍼의 데이터를 상기 로컬 공유 메모리 영역으로 복사하여 상기 원격 메모리들과 데이터를 동기화하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치.

청구항 5

청구항 2에 있어서,

상기 공유 메모리 접근 관리부는

두 개 이상의 공유 메모리 버퍼들 사이의 데이터 누적 연산을 수행하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치.

청구항 6

청구항 5에 있어서,

상기 공유 메모리 접근 관리부는

상기 공유 메모리 서버들에 누적 연산을 요청하고, 상기 공유 메모리 서버들로부터 상기 공유 메모리 버퍼 세그먼트들에 대하여 누적 연산을 수행한 결과를 수신하여 상기 데이터 누적 연산을 수행하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치.

청구항 7

청구항 2에 있어서,

상기 공유 메모리 접근 관리부는

상기 공유 메모리 서버들에 상기 공유 메모리 버퍼 세그먼트들의 해제 및 삭제를 요청하고, 상기 공유 메모리 버퍼 세그먼트들의 해제 및 삭제 요청의 결과들을 수신함에 따라 상기 로컬 공유 메모리 영역을 해제 및 삭제하여 상기 공유 메모리 버퍼의 사용을 종료하고,

상기 메모리 맵핑 테이블 관리부는

상기 공유 메모리 버퍼의 사용이 종료되면 상기 메모리 맵핑 테이블을 삭제하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치.

청구항 8

원격 직접 메모리 접근을 통하여 복수의 분산 처리 장치들과 데이터를 송수신하는 통신부;

상기 분산 처리 장치들이 직접 접근할 수 있는 메모리;

동일한 공유 메모리 서버 클러스터를 구성하는 다른 공유 메모리 서버들과 공유 메모리 버퍼를 구성하는 공유 메모리 관리부를 포함하고,

상기 공유 메모리 버퍼는 상기 공유 메모리 서버들로부터 할당된 공유 메모리 버퍼 세그먼트들로 구성되고,

상기 공유 메모리 버퍼는 상기 분산 처리 장치들 각각에 대하여 상기 공유 메모리 버퍼와 동일한 크기로 할당된 로컬 공유 메모리 영역과 메모리 맵핑 테이블을 이용하여 동기화되고,

상기 공유 메모리 버퍼는 복수의 분산 처리 장치들에 의하여 공유되는 가상의 연속된 버퍼에 상응하고,

상기 로컬 공유 메모리 영역은 가상주소만을 반환하는 것이 아니고 실제 물리 메모리에 할당되는 것을 특징으로 하는, 공유 메모리 서버.

청구항 9

삭제

청구항 10

청구항 8에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치로부터 상기 공유 메모리 버퍼를 생성하기 위한 공유 메모리 버퍼 세그먼트의 크기 정보와 함께 상기 공유 메모리 버퍼 세그먼트의 생성 및 할당을 요청을 수신하고, 상기 공유 메모리 버퍼 세그먼트를 생성 및 할당하여 상기 공유 메모리 버퍼를 구성하는 것을 특징으로 하는, 공유 메모리 서버.

청구항 11

청구항 8에 있어서,

상기 공유 메모리 버퍼는

연산에 의하여 특정 로컬 공유 메모리 영역의 데이터가 변경된 경우에 상기 변경된 로컬 공유 메모리 영역의 데이터와 동기화되고, 변경된 데이터로 나머지 로컬 공유 메모리 영역들과 동기화되는 것을 특징으로 하는, 공유 메모리 서버.

청구항 12

청구항 8에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치로부터 두 개 이상의 공유 메모리 버퍼들 사이의 데이터 누적 연산 요청을 수신하고, 상기 데이터 누적 연산의 대상이 되는 공유 메모리 버퍼 세그먼트들에 대하여 누적 연산을 수행하고, 결과를 상기 분산 처리 장치에 반환하는 것을 특징으로 하는, 공유 메모리 서버.

청구항 13

청구항 8에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치가 상기 공유 메모리 버퍼의 사용을 종료하기 위하여 전송한 상기 공유 메모리 버퍼 세그먼트의 해제 및 삭제 요청을 수신하여 상기 공유 메모리 버퍼 세그먼트를 해제 및 삭제하고, 결과를 상기 분산 처리 장치에 반환하여 상기 분산 처리 장치가 상기 로컬 공유 메모리 영역을 해제 및 삭제하고 상기 메모리 맵핑 테이블을 삭제하도록 하는 것을 특징으로 하는, 공유 메모리 서버.

청구항 14

공유 메모리 서버 클러스터를 구성하는 복수의 공유 메모리 서버들에 구비된 원격 메모리들로부터 할당된 공유 메모리 버퍼 세그먼트들로 구성된 공유 메모리 버퍼에 대하여, 로컬 메모리에 상기 공유 메모리 버퍼와 동일한 크기만큼 로컬 공유 메모리 영역을 할당하는 단계;

상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역 사이의 메모리 맵핑 테이블을 관리하는 단계;

상기 원격 메모리들에 직접 접근하여 데이터를 송수신하여 상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역을 동기화하는 단계; 및

상기 로컬 공유 메모리 영역에 대한 주어진 연산을 수행하는 단계를 포함하고,

상기 공유 메모리 버퍼 세그먼트들은 서로 다른 상기 공유 메모리 서버들에 의해 할당되고,

상기 공유 메모리 버퍼는 복수의 분산 처리 장치들에 의하여 공유되는 가상의 연속된 버퍼에 상응하고,

상기 로컬 공유 메모리 영역은 가상주소만을 반환하는 것이 아니고 실제 물리 메모리에 할당되는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 방법.

청구항 15

청구항 14에 있어서,

직접 상기 공유 메모리 버퍼를 생성하여 동일한 분산 처리 프레임워크를 구성하는 다른 분산 처리 장치들에 공유 메모리 버퍼 정보를 공유하거나, 상기 다른 분산 처리 장치에 의하여 생성된 상기 공유 메모리 버퍼에 상응하는 공유 메모리 버퍼 정보를 수신하여 상기 공유 메모리 버퍼 정보를 획득하는 단계

를 더 포함하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 방법.

청구항 16

청구항 15에 있어서,

상기 공유 메모리 버퍼 정보를 획득하는 단계는

상기 공유 메모리 서버들에 각각에 상응하는 상기 공유 메모리 버퍼 세그먼트들의 크기를 계산하는 단계; 및

상기 공유 메모리 서버들에 상기 공유 메모리 버퍼 세그먼트들의 생성 및 할당을 요청하여 상기 공유 메모리 버퍼를 생성하는 단계

를 포함하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 방법.

청구항 17

청구항 15에 있어서,

상기 동기화하는 단계는

상기 연산에 의하여 상기 로컬 공유 메모리 영역의 데이터가 변경된 경우에 상기 로컬 공유 메모리 영역의 데이터를 상기 공유 메모리 버퍼에 복사하여 상기 원격 메모리들과 데이터를 동기화하고, 상기 다른 분산 처리 장치들에 의하여 상기 공유 메모리 버퍼의 데이터가 변경된 경우에 상기 공유 메모리 버퍼의 데이터를 상기 로컬 공유 메모리 영역으로 복사하여 상기 원격 메모리들과 데이터를 동기화하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 방법.

청구항 18

청구항 15에 있어서,

두 개 이상의 공유 메모리 버퍼들 사이의 데이터 누적 연산을 수행하는 단계

를 더 포함하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 방법.

청구항 19

청구항 18에 있어서,

상기 데이터 누적 연산을 수행하는 단계는

상기 공유 메모리 서버들에 누적 연산을 요청하는 단계; 및

상기 공유 메모리 서버들로부터 상기 공유 메모리 버퍼 세그먼트들에 대하여 누적 연산을 수행한 결과를 수신하는 단계

를 포함하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 방법.

청구항 20

청구항 15에 있어서,

상기 공유 메모리 서버들에 상기 공유 메모리 버퍼 세그먼트들의 해제 및 삭제를 요청하고, 상기 공유 메모리 버퍼 세그먼트들의 해제 및 삭제 요청의 결과들을 수신함에 따라 상기 로컬 공유 메모리 영역을 해제 및 삭제하고, 상기 메모리 맵핑 테이블을 삭제하여 상기 공유 메모리 버퍼의 사용을 종료하는 단계

를 더 포함하는 것을 특징으로 하는 하는, 원격 직접 메모리 접근을 통한 분산 처리 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 원격 직접 메모리 접근을 통한 분산 처리 장치 및 그 방법에 관한 것으로, 구체적으로 분산 처리 프레임워크에서 공유 메모리 서버들의 메모리들로 구성된 가상의 공유 메모리 버퍼에 직접 접근하는 분산 처리 장치 및 그 방법에 관한 것이다.

배 경 기 술

[0002] 분산 병렬 처리란 다수의 계산 자원을 동시에 병렬로 사용하여 대규모 데이터 분석을 빠르게 수행하는 것이다.

다수의 계산 노드에 분산 병렬 실행되는 프로세스들은 상호간에 데이터 공유가 필수적이며 대표적인 데이터 공유 방식으로 MPI(Message Passing Interface)를 들 수 있다. 그러나 분산 처리 데이터의 일부만을 일부 프로세스 간에 메시지 패싱 형태로 전달하는 형태가 아니라 지속적으로 전체 분산 처리 프로세스 간에 전체 처리 데이터를 비동기적으로 업데이트하고 참조하는 경우에는 MPI 방식보다는 공유 메모리 형태로 공유하는 것이 더 유리하다.

[0003] 분산 처리 플랫폼에서 분산 처리를 수행하는 프로세스들은 상호 간에 대규모 공유 데이터를 빈번하게 송수신해야 하며, 이에 따른 통신 오버헤드는 전체 분산 처리 성능이나 처리 시간에서 차지하는 비중이 매우 높다. 통신 오버헤드가 높을수록 계산 노드의 계산 프로세서(예컨대, CPU, GPU 등)들은 대기하는 시간이 길어지고 이는 자원 사용률 저하로 나타난다. 통신 오버헤드가 높은 이유는 TCP/IP를 포함한 대부분의 통신 프로토콜 스택은 응용 프로세스가 보내는 메시지를 다단계의 프로토콜 레이어를 통해 처리하는 프로토콜 처리 오버헤드와 프로토콜 처리중에 1회 이상 메모리 복사가 발생 때문이다. 따라서, 원격 직접 메모리 접근(RDMA: Remote Direct Memory Access)을 통하여 분산 처리에 따른 통신 오버헤드를 줄이는 것이 요구된다.

[0004] 전술한 배경기술은 발명자가 본 발명의 도출을 위해 보유하고 있었거나, 본 발명의 도출 과정에서 습득한 기술정보로서, 반드시 본 발명의 출원 전에 일반 공중에게 공개된 공지기술이라할 수는 없다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 국내 공개특허공보 제10-2006-0009244호

발명의 내용

해결하려는 과제

[0006] 본 발명의 목적은 원격 직접 메모리 접근을 통하여 분산 처리 데이터를 공유하는 원격 직접 메모리 접근을 통한 분산 처리 장치 및 그 방법을 제공하는 것이다.

[0007] 또한, 본 발명의 목적은 다수의 공유 메모리 서버들을 클러스터링하고 각각의 공유 메모리 서버들로부터 공유 메모리 버퍼 세그먼트들을 할당하여 공유 메모리 버퍼를 구성하고, 공유 메모리 버퍼에 원격 직접 메모리 접근하는 분산 처리 장치 및 그 방법을 제공하는 것이다.

과제의 해결 수단

[0008] 본 발명의 일 실시예는, 공유 메모리 서버 클러스터를 구성하는 복수의 공유 메모리 서버들에 구비된 원격 메모리들에 직접 접근하여 데이터를 송수신하는 통신부; 상기 원격 메모리들로부터 할당된 공유 메모리 버퍼 세그먼트들로 구성된 공유 메모리 버퍼에 대하여, 메모리에 상기 공유 메모리 버퍼와 동일한 크기만큼 로컬 공유 메모리 영역을 할당하고, 상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역을 동기화하는 공유 메모리 접근 관리부; 상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역 사이의 메모리 맵핑 테이블을 관리하는 메모리 맵핑 테이블 관리부; 및 상기 로컬 공유 메모리 영역에 대한 주어진 연산을 수행하는 연산부를 포함하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 장치를 제공한다.

[0009] 이때, 상기 공유 메모리 접근 관리부는 직접 상기 공유 메모리 버퍼를 생성하여 동일한 분산 처리 프레임워크를 구성하는 다른 분산 처리 장치들에 공유 메모리 버퍼 정보를 공유하거나, 상기 다른 분산 처리 장치에 의하여 생성된 상기 공유 메모리 버퍼에 상응하는 공유 메모리 버퍼 정보를 수신할 수 있다.

[0010] 이때, 상기 공유 메모리 접근 관리부는 상기 공유 메모리 서버들에 각각에 상응하는 상기 공유 메모리 버퍼 세그먼트들의 크기를 계산하고, 상기 공유 메모리 서버들에 상기 공유 메모리 버퍼 세그먼트들의 생성 및 할당을 요청하여 상기 공유 메모리 버퍼를 생성할 수 있다.

[0011] 이때, 상기 공유 메모리 접근 관리부는 상기 연산에 의하여 상기 로컬 공유 메모리 영역의 데이터가 변경된 경우에 상기 로컬 공유 메모리 영역의 데이터를 상기 공유 메모리 버퍼에 복사하여 상기 원격 메모리들과 데이터를 동기화하고, 상기 다른 분산 처리 장치들에 의하여 상기 공유 메모리 버퍼의 데이터가 변경된 경우에 상기 공유 메모리 버퍼의 데이터를 상기 로컬 공유 메모리 영역으로 복사하여 상기 원격 메모리들과 데이터를 동기화

할 수 있다.

[0012] 이때, 상기 공유 메모리 접근 관리부는 두 개 이상의 공유 메모리 버퍼들 사이의 데이터 누적 연산을 수행할 수 있다.

[0013] 이때, 상기 공유 메모리 접근 관리부는 상기 공유 메모리 서버들에 누적 연산을 요청하고, 상기 공유 메모리 서버들로부터 상기 공유 메모리 버퍼 세그먼트들에 대하여 누적 연산을 수행한 결과를 수신하여 상기 데이터 누적 연산을 수행할 수 있다.

[0014] 이때, 상기 공유 메모리 접근 관리부는 상기 공유 메모리 서버들에 상기 공유 메모리 버퍼 세그먼트들의 해제 및 삭제를 요청하고, 상기 공유 메모리 버퍼 세그먼트들의 해제 및 삭제 요청의 결과들을 수신함에 따라 상기 로컬 공유 메모리 영역을 해제 및 삭제하여 상기 공유 메모리 버퍼의 사용을 종료하고, 상기 메모리 맵핑 테이블 관리부는 상기 공유 메모리 버퍼의 사용이 종료되면 상기 메모리 맵핑 테이블을 삭제할 수 있다.

[0015] 본 발명의 다른 일 실시예는, 원격 직접 메모리 접근을 통하여 복수의 분산 처리 장치들과 데이터를 송수신하는 통신부; 상기 분산 처리 장치들이 직접 접근할 수 있는 메모리; 동일한 공유 메모리 서버 클러스터를 구성하는 다른 공유 메모리 서버들과 공유 메모리 버퍼를 구성하는 공유 메모리 관리부를 포함하는 것을 특징으로 하는, 공유 메모리 서버를 제공한다.

[0016] 이때, 상기 공유 메모리 버퍼는 상기 분산 처리 장치들 각각에 대하여 상기 공유 메모리 버퍼와 동일한 크기로 할당된 로컬 공유 메모리 영역과 메모리 맵핑 테이블을 이용하여 동기화될 수 있다.

[0017] 이때, 상기 공유 메모리 관리부는 상기 분산 처리 장치로부터 상기 공유 메모리 버퍼를 생성하기 위한 공유 메모리 버퍼 세그먼트의 크기 정보와 함께 상기 공유 메모리 버퍼 세그먼트의 생성 및 할당을 요청을 수신하고, 상기 공유 메모리 버퍼 세그먼트를 생성 및 할당하여 상기 공유 메모리 버퍼를 구성할 수 있다.

[0018] 이때, 상기 공유 메모리 버퍼는 연산에 의하여 특정 로컬 공유 메모리 영역의 데이터가 변경된 경우에 상기 변경된 로컬 공유 메모리 영역의 데이터와 동기화되고, 변경된 데이터로 나머지 로컬 공유 메모리 영역들과 동기화될 수 있다.

[0019] 이때, 상기 공유 메모리 관리부는 상기 분산 처리 장치로부터 두 개 이상의 공유 메모리 버퍼들 사이의 데이터 누적 연산 요청을 수신하고, 상기 데이터 누적 연산의 대상이 되는 공유 메모리 버퍼 세그먼트들에 대하여 누적 연산을 수행하고, 결과를 상기 분산 처리 장치에 반환할 수 있다.

[0020] 이때, 상기 공유 메모리 관리부는 상기 분산 처리 장치가 상기 공유 메모리 버퍼의 사용을 종료하기 위하여 전송한 상기 공유 메모리 버퍼 세그먼트의 해제 및 삭제 요청을 수신하여 상기 공유 메모리 버퍼 세그먼트를 해제 및 삭제하고, 결과를 상기 분산 처리 장치에 반환하여 상기 분산 처리 장치가 상기 로컬 공유 메모리 영역을 해제 및 삭제하고 상기 메모리 맵핑 테이블을 삭제하도록 할 수 있다.

[0021] 본 발명의 다른 일 실시예는, 공유 메모리 서버 클러스터를 구성하는 복수의 공유 메모리 서버들에 구비된 원격 메모리들로부터 할당된 공유 메모리 버퍼 세그먼트들로 구성된 공유 메모리 버퍼에 대하여, 메모리에 상기 공유 메모리 버퍼와 동일한 크기만큼 로컬 공유 메모리 영역을 할당하는 단계; 상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역 사이의 메모리 맵핑 테이블을 관리하는 단계; 상기 원격 메모리들에 직접 접근하여 데이터를 송수신하여 상기 공유 메모리 버퍼와 상기 로컬 공유 메모리 영역을 동기화하는 단계; 및 상기 로컬 공유 메모리 영역에 대한 주어진 연산을 수행하는 단계를 포함하는 것을 특징으로 하는, 원격 직접 메모리 접근을 통한 분산 처리 방법을 제공한다.

[0022] 이때, 직접 상기 공유 메모리 버퍼를 생성하여 동일한 분산 처리 프레임워크를 구성하는 다른 분산 처리 장치들에 공유 메모리 버퍼 정보를 공유하거나, 상기 다른 분산 처리 장치에 의하여 생성된 상기 공유 메모리 버퍼에 상응하는 공유 메모리 버퍼 정보를 수신하여 상기 공유 메모리 버퍼 정보를 획득하는 단계를 더 포함할 수 있다.

[0023] 이때, 상기 공유 메모리 버퍼 정보를 획득하는 단계는 상기 공유 메모리 서버들에 각각에 상응하는 상기 공유 메모리 버퍼 세그먼트들의 크기를 계산하는 단계; 및 상기 공유 메모리 서버들에 상기 공유 메모리 버퍼 세그먼트들의 생성 및 할당을 요청하여 상기 공유 메모리 버퍼를 생성하는 단계를 포함할 수 있다.

[0024] 이때, 상기 동기화하는 단계는 상기 연산에 의하여 상기 로컬 공유 메모리 영역의 데이터가 변경된 경우에 상기 로컬 공유 메모리 영역의 데이터를 상기 공유 메모리 버퍼에 복사하여 상기 원격 메모리들과 데이터를 동기화하

고, 상기 다른 분산 처리 장치들에 의하여 상기 공유 메모리 버퍼의 데이터가 변경된 경우에 상기 공유 메모리 버퍼의 데이터를 상기 로컬 공유 메모리 영역으로 복사하여 상기 원격 메모리들과 데이터를 동기화할 수 있다.

[0025] 이때, 두 개 이상의 공유 메모리 버퍼들 사이의 데이터 누적 연산을 수행하는 단계를 더 포함할 수 있다.

[0026] 이때, 상기 데이터 누적 연산을 수행하는 단계는 상기 공유 메모리 서버들에 누적 연산을 요청하는 단계; 및 상기 공유 메모리 서버들로부터 상기 공유 메모리 버퍼 세그먼트들에 대하여 누적 연산을 수행한 결과를 수신하는 단계를 포함할 수 있다.

[0027] 이때, 상기 공유 메모리 서버들에 상기 공유 메모리 버퍼 세그먼트들의 해제 및 삭제를 요청하고, 상기 공유 메모리 버퍼 세그먼트들의 해제 및 삭제 요청의 결과들을 수신함에 따라 상기 로컬 공유 메모리 영역을 해제 및 삭제하고, 상기 메모리 맵핑 테이블을 삭제하여 상기 공유 메모리 버퍼의 사용을 종료하는 단계를 더 포함할 수 있다.

발명의 효과

[0028] 본 발명에 따르면, 원격 직접 메모리 접근을 통한 분산 처리 장치 및 그 방법에 의해, 원격 직접 메모리 접근을 통하여 분산 처리 데이터를 공유함으로써 분산 처리시에 발생하는 통신 오버로드를 효과적으로 낮출 수 있다.

[0029] 또한, 본 발명은 원격 직접 메모리 접근을 통한 분산 처리 장치 및 그 방법에 의해, 다수의 공유 메모리 서버들을 클러스터링하고 각각의 공유 메모리 서버들로부터 공유 메모리 버퍼 세그먼트들을 할당하여 공유 메모리 버퍼를 구성하고 분산 처리 장치가 공유 메모리 버퍼에 원격 직접 메모리 접근함으로써, 공유 메모리 서버들 사이의 별도의 동기화 작업이 없이 효율적으로 메모리 데이터를 관리할 수 있다.

도면의 간단한 설명

[0030] 도 1은 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 시스템의 구성을 나타낸 도면이다.

도 2는 도 1에 도시된 원격 직접 메모리 접근을 통한 분산 처리 장치의 일 예를 나타낸 블록도이다.

도 3은 도 1에 도시된 공유 메모리 서버의 일 예를 나타낸 블록도이다.

도 4는 본 발명의 일 실시예에 따른 공유 메모리 버퍼를 구성하는 방법을 나타낸 도면이다.

도 5는 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 방법을 나타낸 동작 흐름도이다.

도 6은 도 5에 도시된 공유 메모리 버퍼를 생성 및 할당하는 단계의 일 예를 나타낸 동작 흐름도이다.

도 7은 도 5에 도시된 공유 메모리 버퍼를 해제 및 삭제하는 단계의 일 예를 나타낸 동작 흐름도이다.

도 8은 본 발명의 일 실시예에 따른 공유 메모리 버퍼들의 데이터 누적 연산 방법을 나타낸 동작이다.

도 9는 도 8에 도시된 공유 메모리 버퍼들의 데이터 누적 연산 방법을 나타낸 동작 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0031] 본 발명은 다양한 변환을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 본 발명의 효과 및 특징, 그리고 그것들을 달성하는 방법은 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 여기서, 반복되는 설명, 본 발명의 요지를 불필요하게 흐릴 수 있는 공지 기능, 및 구성에 대한 상세한 설명은 생략한다. 본 발명의 실시형태는 당 업계에서 평균적인 지식을 가진 자에게 본 발명을 보다 완전하게 설명하기 위해서 제공되는 것이다. 따라서, 도면에서의 요소들의 형상 및 크기 등은 보다 명확한 설명을 위해 과장될 수 있다.

[0032] 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성되어 다양한 형태로 구현될 수 있다. 이하의 실시예에서, 제1, 제2 등의 용어는 한정적인 의미가 아니라 하나의 구성 요소를 다른 구성 요소와 구별하는 목적으로 사용되었다. 또한, 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는한 복수의 표현을 포함한다. 또한, 포함하다 또는 가지다 등의 용어는 명세서상에 기재된 특징, 또는 구성요소가 존재함을 의미하는 것이고, 하나 이상의 다른 특징들 또는 구성요소가 부가될 가능성을 미리 배제하는 것은 아니다.

[0033] 이하, 첨부된 도면을 참조하여 본 발명의 실시예들을 상세히 설명하기로 하며, 도면을 참조하여 설명할 때 동일하거나 대응하는 구성 요소는 동일한 도면 부호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다.

[0035] 도 1은 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 시스템(100)의 구성을 나타낸 도면이다.

[0036] 도 1을 참조하면, 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 시스템(100)에서 복수의 원격 직접 메모리 접근을 통한 분산 처리 장치들(110)은 원격 직접 메모리 접근(RDMA: Remote Direct Memory Access) 지원 네트워크(130)을 통해 복수의 공유 메모리 서버들(120)과 상호 연결된다. 여기서, 공유 메모리 서버들(120)은 하나의 공유 메모리 서버 클러스터(140)를 구성한다.

[0037] 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 장치(110)는 공유 메모리 서버 클러스터를 구성하는 복수의 공유 메모리 서버들에 구비된 원격 메모리들로부터 할당된 공유 메모리 버퍼 세그먼트들로 구성된 공유 메모리 버퍼에 대하여, 메모리에 공유 메모리 버퍼와 동일한 크기만큼 로컬 공유 메모리 영역을 할당하고, 공유 메모리 버퍼와 로컬 공유 메모리 영역 사이의 메모리 맵핑 테이블을 이용하여 공유 메모리 버퍼와 로컬 공유 메모리 영역을 동기화하는 것을 특징으로 한다. 그리고, 로컬 공유 메모리 영역에 대하여 주어진 혹은 입력된 연산을 처리한다.

[0038] 여기서, 복수의 원격 직접 메모리 접근을 통한 분산 처리 장치들(110)은 하나의 분산 처리 프레임워크에 포함될 수 있다. 또한, 하나 이상의 원격 직접 메모리 접근을 통한 분산 처리 장치들(110)은 하나의 계산 노드로 구성되어 연산 기능을 제공할 수 있다.

[0039] 이때, 원격 직접 메모리 접근을 통한 분산 처리 장치(110)는 분산 처리 프레임워크에서 주도적으로 분산 처리 작업의 초기화 및 제어를 처리하는 마스터 분산 처리 장치 또는 마스터 분산 처리 장치의 제어를 받으며 계산을 담당하는 슬레이브 분산 처리 장치 혹은 워커 분산 처리 장치로 구분될 수 있다.

[0040] 이때, 마스터 분산 처리 장치는 공유 메모리 서버들(120)에 데이터를 저장하기 위한 공유 메모리 버퍼를 생성하고 슬레이브 분산 처리 장치들에 공유 메모리 버퍼 정보를 전달하여 모든 분산 처리 장치들(110)이 공유 메모리 서버(120)상의 동일 메모리 세그먼트 영역을 접근할 수 있도록 한다. 여기서, 공유 메모리 버퍼 정보에는 공유 메모리 버퍼 전체 크기, 공유 메모리 버퍼 생성키, 공유 메모리 서버별로 생성된 공유 메모리 버퍼 세그먼트 정보 등이 포함될 수 있다.

[0041] 이때, 원격 직접 메모리 접근을 통한 분산 처리 장치(110)는 연산에 의하여 로컬 공유 메모리 영역의 데이터가 변경된 경우에 로컬 공유 메모리 영역의 데이터를 공유 메모리 버퍼에 복사하여 원격 메모리들과 데이터를 동기화할 수 있다. 또한, 다른 원격 직접 메모리 접근을 통한 분산 처리 장치(110)의 연산에 의하여 공유 메모리 버퍼의 데이터가 변경된 경우에 공유 메모리 버퍼의 데이터를 로컬 공유 메모리 영역으로 복사하여 원격 메모리들과 데이터를 동기화할 수 있다.

[0042] 공유 메모리 서버(120)는 분산 처리 프레임워크에서 공유 메모리를 제공하는 장치이다.

[0043] 여기서, 복수의 공유 메모리 서버들(120)은 하나의 공유 메모리 서버 클러스터를 구성할 수 있다. 또한, 하나 이상의 공유 메모리 서버들(110)은 하나의 메모리 서비스 노드로 구성되어 공유 메모리 서비스를 제공할 수 있다.

[0044] 이때, 복수의 공유 메모리 서버들(120)은 각각 공유 메모리 버퍼의 생성 및 할당 요청에 따라 공유 메모리 버퍼 세그먼트를 생성 및 할당하여, 각각의 공유 메모리 버퍼 세그먼트들을 연결한 가상의 공유 메모리 버퍼를 제공할 수 있다. 여기서, 공유 메모리 버퍼는 분산 처리 장치(110)가 원격 직접 메모리 접근 지원 네트워크(130)를 통하여 직접 접근할 수 있다.

[0045] 이때, 공유 메모리 버퍼는 분산 처리 장치들(110)의 로컬 공유 메모리 영역과 동기화될 수 있다. 즉, RDMA 읽기/쓰기를 통하여 동기화할 수 있다.

[0046] 이때, 공유 메모리 서버(120)는 공유 메모리 버퍼 세그먼트들 간의 누적 연산 기능을 제공할 수 있다.

[0047] 원격 직접 메모리 접근 지원 네트워크(130)는 복수의 분산 처리 장치들(110)과 복수의 공유 메모리 서버들(120) 사이의 통신을 제공하는 네트워크로, 분산 처리 장치들(110)이 공유 메모리 서버들(120)의 메모리에 직접 접근 가능한 기능을 제공한다.

[0048] 즉, 원격 직접 메모리 접근을 지원하는 고성능 네트워크(130)로 연결된 고성능 컴퓨팅 클러스터 시스템 환경에

서 분산 처리 장치들(110)이 다수의 공유 메모리 서버들(120)의 물리 메모리 세그먼트들을 결합하여 가상의 연속된 공유 메모리 버퍼에 직접 접근할 수 있도록 함으로써, 분산 처리 장치들 간의 데이터 공유를 가속화하고 효율성을 높일 수 있다.

[0049] 이와 같은 공유 메모리 형태로 분산 처리 데이터를 공유하는 대표적인 분산 처리 방식으로는 비동기 딥러닝 트레이닝에 이용될 수 있다. 비동기 딥러닝 트레이닝 방식은 데이터 병렬 딥러닝 학습 방식의 하나로, 학습 데이터를 나누어 다수의 딥러닝 프로세스가 학습을 수행하고, 학습하는 도중에 학습한 내용을 다른 딥러닝 프로세스들과 공유 데이터 버퍼를 통해 비동기적으로 공유하는 학습 방법이다. 비동기 딥러닝 트레이닝 방식에서 각 딥러닝 분산 처리 프로세스는 딥러닝 파라미터(딥러닝 트레이닝에서 트레이닝의 대상이 되는 가중치와 특징값의 총칭)를 다른 프로세스들과 동기를 맞추지 않고 비동기적으로 파라미터를 업데이트하는데, 이 방식은 본 발명에서 제안하는 공유 메모리 구조에 적합하다.

[0050] 또한, 본 발명에서 제안하는 방식은 파라미터 서버와 일부 유사하나, 딥러닝 분산 프로세스들로부터 그래디언트를 받아 능동적으로 가중치 파라미터를 계산하여 직접 딥러닝 파라미터를 업데이트하는 파라미터 서버 방식과 달리 분산 처리 장치들이 원격 직접 메모리 접근 기능을 통해 공유 메모리 서버의 개입 없이 공유 메모리 버퍼를 직접 읽고 쓰는 것이 가능하다.

[0051] 또한, 하나의 단일 메모리 서버상의 메모리만을 공유 메모리로 사용할 경우에는 확장성에 제한이 있으나, 공유 메모리 서버 클러스터를 구성함으로써 대규모 분산 처리에도 유연한 확장성을 제공할 수 있다.

[0053] 도 2는 도 1에 도시된 원격 직접 메모리 접근을 통한 분산 처리 장치(110)의 일 예를 나타낸 블록도이다.

[0054] 도 2를 참조하면, 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 장치(110)는 제어부(210), 통신부(220), 메모리(230), 연산 처리부(240) 및 공유 메모리 서버 접근 지원부(250) 등을 포함한다.

[0055] 상세히, 제어부(210)는 일종의 중앙처리장치로서 원격 직접 메모리 접근을 통한 분산 처리 과정을 제어한다. 즉, 제어부(210)는 메모리(230), 연산 처리부(240) 및 공유 메모리 서버 접근 지원부(250) 등을 제어하여 다양한 기능을 제공할 수 있다.

[0056] 여기서, 제어부(210)는 프로세서(processor)와 같이 데이터를 처리할 수 있는 모든 종류의 장치를 포함할 수 있다. 여기서, '프로세서(processor)'는, 예를 들어 프로그램 내에 포함된 코드 또는 명령으로 표현된 기능을 수행하기 위해 물리적으로 구조화된 회로를 갖는, 하드웨어에 내장된 데이터 처리 장치를 의미할 수 있다. 이와 같이 하드웨어에 내장된 데이터 처리 장치의 일 예로써, 마이크로프로세서(microprocessor), 중앙처리장치(central processing unit: CPU), 프로세서 코어(processor core), 멀티프로세서(multiprocessor), ASIC(application-specific integrated circuit), FPGA(field programmable gate array) 등의 처리 장치를 망라할 수 있으나, 본 발명의 범위가 이에 한정되는 것은 아니다.

[0057] 통신부(220)는 RDMA 지원 네트워크(도 1의 130 참조)를 통하여 원격 직접 메모리 접근을 통한 분산 처리 장치(110)와 공유 메모리 서버(도 1의 120 참조) 간의 송수신 신호를 전송하는데 필요한 통신 인터페이스를 제공한다.

[0058] 여기서, 통신부(220)는 다른 네트워크 장치와 유무선 연결을 통해 제어 신호 또는 데이터 신호와 같은 신호를 송수신하기 위해 필요한 하드웨어 및 소프트웨어를 포함하는 장치일 수 있다.

[0059] 이때, 통신부(220)는 RDMA 지원 네트워크(도 1의 130 참조)를 통해 공유 메모리 서버들(도 1의 120 참조)의 원격 메모리들에 직접 접근하여 데이터를 읽고 쓸 수 있다.

[0060] 메모리(230)는 제어부(210)가 처리하는 데이터를 일시적 또는 영구적으로 저장하는 기능을 수행한다. 여기서, 메모리(230)는 자기 저장 매체(magnetic storage media) 또는 플래시 저장 매체(flash storage media)를 포함할 수 있으나, 본 발명의 범위가 이에 한정되는 것은 아니다.

[0061] 이때, 메모리(230)는 공유 메모리 서버들(도 1의 120 참조)의 원격 메모리들로부터 구성된 공유 메모리 버퍼와 동일한 크기만큼을 로컬 공유 메모리 영역으로 할당하고, 공유 메모리 버퍼와 동기화할 수 있다.

[0062] 이에 따라, 분산 처리 장치(110)는 로컬 공유 메모리 영역에 대하여 연산을 수행하고, 이를 공유 메모리 버퍼와 동기화함으로써 다른 분산 처리 장치들과 메모리를 공유할 수 있다.

[0063] 연산 처리부(240)는 분산 처리 프레임워크에서 분산 처리 장치(110)에 주어진 연산을 수행한다.

[0064] 이때, 연산 처리부(240)는 로컬 공유 메모리 영역에 대하여 연산을 수행할 수 있다.

[0065] 이때, 연산 처리부(240)는 API(Application Programmable Interface)를 통해 공유 메모리 서버 접근 지원부(250)에 명시적으로 로컬 공유 메모리 영역과 공유 메모리 버퍼의 동기화를 요청 및 수행할 수 있다.

[0066] 이때, 연산 처리부(240)는 API(Application Programmable Interface)를 통해 공유 메모리 서버 접근 지원부(250)에 복수의 공유 메모리 버퍼들에 대한 누적 연산을 요청 및 수행할 수 있다.

[0067] 공유 메모리 서버 접근 지원부(250)는 공유 메모리 서버(도 1의 120 참조)와의 RDMA 읽기/쓰기를 통한 접근을 지원하여, 공유 메모리 버퍼와의 동기화를 지원한다.

[0068] 이때, 공유 메모리 서버 접근 지원부(250)는 공유 메모리 서버 클러스터를 등록하여 공유 메모리 서버 클러스터를 구성하는 공유 메모리 서버들(도 1의 120 참조)의 정보를 획득할 수 있다. 여기서, 공유 메모리 서버 클러스터의 등록은 사용자가 입력한 공유 메모리 서버들(도 1의 120 참조)의 접근 정보를 이용하여 해당 공유 메모리 서버들(도 1의 120 참조)과의 연결을 설정하고 공유 메모리 버퍼 세그먼트를 생성 및 할당받는 초기화 과정을 의미할 수 있다. 그리고, 공유 메모리 서버 접근 정보에는 IP 주소와 포트 번호 정보 등이 포함될 수 있다. 특히, 공유 메모리 서버 클러스터를 등록할 때, 모든 분산 처리 장치들(110)은 모든 공유 메모리 서버들(도 1의 120 참조)의 순서를 동일하게 등록할 수 있다.

[0069] 이때, 공유 메모리 서버 접근 지원부(250)는 공유 메모리 서버 클러스터에 등록된 각 공유 메모리 서버들(도 1의 120 참조)에 대하여 공유 메모리 버퍼를 구성하는 공유 메모리 버퍼 세그먼트들의 크기를 계산할 수 있다. 예컨대, 하나의 공유 메모리 서버 클러스터에 5개의 공유 메모리 서버들이 포함되어 있고, 공유 메모리 버퍼의 크기를 5GB로 구성하는 경우, 5개의 크기 1GB의 공유 메모리 버퍼 세그먼트들로 나눌 수 있다. 여기서, 각 공유 메모리 버퍼 세그먼트들의 크기는 동일하지 않을 수 있다.

[0070] 이때, 공유 메모리 서버 접근 지원부(250)는 공유 메모리 서버들(도 1의 120 참조)에 계산된 공유 메모리 서버 세그먼트들의 크기 정보와 공유 메모리 버퍼 세그먼트의 생성 및 할당을 요청을 전달하여 공유 메모리 버퍼를 구성할 수 있다.

[0071] 이때, 공유 메모리 서버 접근 지원부(250)는 다른 분산 처리 장치가 구성한 공유 메모리 버퍼에 대한 접근권을 획득하여 공유 메모리 버퍼의 생성 및 할당을 대신할 수 있다.

[0072] 이때, 공유 메모리 서버 접근 지원부(250)는 공유 메모리 버퍼를 구성하기 위하여 공유 메모리 서버들(도 1의 120 참조)에 공유 메모리 버퍼 생성키를 함께 전송할 수 있다. 여기서, 공유 메모리 버퍼 생성키는 동일한 공유 메모리 버퍼가 중복 생성을 방지하거나 유효한 요청인지 여부를 확인하거나 공유 메모리 버퍼 세그먼트를 특정하기 위한 목적으로 이용될 수 있다.

[0073] 이때, 공유 메모리 서버 접근 지원부(250)는 모든 공유 메모리 서버들(도 1의 120 참조)에 대하여 공유 메모리 버퍼 세그먼트들의 생성 및 할당이 이루어지면 메모리(230)에 공유 메모리 버퍼와 동일한 크기로 로컬 공유 메모리 영역을 할당할 수 있다. 여기서, 로컬 공유 메모리 영역은 실제 물리 메모리에 할당되며, 로컬 공유 메모리 영역이 할당됨에 따라 주소 정보(예컨대, 가상 주소)가 반환된다.

[0074] 이때, 공유 메모리 서버 접근 지원부(250)는 로컬 공유 메모리 영역을 공유 메모리 버퍼와 동기화를 수행할 수 있다. 여기서, 동기화는 메모리 맵핑 테이블을 이용하여 수행될 수 있다.

[0075] 이때, 공유 메모리 서버 접근 지원부(250)는 연산 처리부(240)에 의하여 로컬 공유 메모리 영역의 데이터가 변경된 경우, RDMA 통해 로컬 공유 메모리 영역의 데이터를 공유 메모리 버퍼에 복사하여 동기화할 수 있다. 또는, 변경된 데이터에 대하여만 복사하여 동기화할 수 있다.

[0076] 이때, 공유 메모리 서버 접근 지원부(250)는 다른 분산 처리 장치의 연산에 의하여 공유 메모리 버퍼의 데이터가 변경된 경우, RDMA를 통해 공유 메모리 버퍼의 데이터를 로컬 공유 메모리 영역에 복사하여 동기화할 수 있다. 또는, 변경된 데이터에 대하여만 복사하여 동기화할 수 있다.

[0077] 이때, 공유 메모리 서버 접근 지원부(250)는 공유 메모리 버퍼의 사용이 종료된 경우 공유 메모리 서버들(도 1의 120 참조)에 공유 메모리 버퍼 세그먼트들의 해제 및 삭제를 요청할 수 있다.

[0078] 이때, 공유 메모리 서버 접근 지원부(250)는 모든 공유 메모리 버퍼 세그먼트들의 해제 및 삭제가 이루어지면, 공유 메모리 서버들(도 1의 120 참조)과의 연결을 종료하고 공유 메모리 서버 클러스터를 등록 해제하며 정보를

삭제하여 공유 메모리 버퍼의 사용을 종료할 수 있다.

[0079] 이때, 공유 메모리 서버 접근 지원부(250)는 복수의 공유 메모리 버퍼에 대하여 데이터 누적 연산 기능을 제공할 수 있다. 예컨대, 제1 공유 메모리 버퍼와 제2 공유 메모리 버퍼에 대한 누적 연산을 수행하는 경우, 제1 공유 메모리 버퍼에 대한 데이터 동기화를 수행하고, 각 공유 메모리 서버들(도 1의 120 참조)에 제1 공유 버퍼 세그먼트들로부터 제2 공유 버퍼 세그먼트들로의 누적 연산을 요청하고, 모든 공유 메모리 서버들(도 1의 120 참조)에서 누적 연산이 완료되면 그 결과를 반환할 수 있다. 각 공유 메모리 서버들(도 1의 120 참조)에서는 누적 연산을 위하여 제2 공유 메모리 버퍼 세그먼트를 잠그고, 제1 공유 메모리 버퍼 세그먼트에서 제2 공유 메모리 버퍼 세그먼트로의 누적 연산을 수행할 수 있다.

[0080] 메모리 맵핑 테이블 관리부(260)는 로컬 공유 메모리 영역과 공유 메모리 버퍼 사이의 메모리 맵핑 테이블을 관리한다.

[0081] 이때, 메모리 맵핑 테이블 관리부(260)는 저장소를 포함하여 직접 메모리 맵핑 테이블을 저장하여 관리할 수도 있지만, 별도의 저장소나 메모리(230)에 메모리 맵핑 테이블을 저장하여 관리할 수 있다.

[0082] 이때, 메모리 맵핑 테이블 관리부(260)는 공유 메모리 버퍼가 생성되면 공유 메모리 버퍼와 로컬 공유 메모리 영역 사이의 메모리 맵핑 테이블을 생성할 수 있다.

[0083] 이때, 메모리 맵핑 테이블 관리부(260)는 공유 메모리 버퍼의 사용이 종료되면 메모리 맵핑 테이블을 삭제할 수 있다.

[0085] 도 3은 도 1에 도시된 공유 메모리 서버(120)의 일 예를 나타낸 블록도이다.

[0086] 도 3을 참조하면, 본 발명의 일 실시예에 따른 공유 메모리 서버(120)는 제어부(310), 통신부(320), 메모리(330) 및 공유 메모리 관리부(340) 등을 포함한다.

[0087] 상세히, 제어부(310)는 일종의 중앙처리장치로서 원격 직접 메모리 접근을 통한 분산 처리 과정을 제어한다. 즉, 제어부(310)는 메모리(330) 및 공유 메모리 관리부(340) 등을 제어하여 다양한 기능을 제공할 수 있다.

[0088] 여기서, 제어부(310)는 프로세서(processor)와 같이 데이터를 처리할 수 있는 모든 종류의 장치를 포함할 수 있다. 여기서, '프로세서(processor)'는, 예를 들어 프로그램 내에 포함된 코드 또는 명령으로 표현된 기능을 수행하기 위해 물리적으로 구조화된 회로를 갖는, 하드웨어에 내장된 데이터 처리 장치를 의미할 수 있다. 이와 같이 하드웨어에 내장된 데이터 처리 장치의 일 예로써, 마이크로프로세서(microprocessor), 중앙처리장치(central processing unit: CPU), 프로세서 코어(processor core), 멀티프로세서(multiprocessor), ASIC(application-specific integrated circuit), FPGA(field programmable gate array) 등의 처리 장치를 망라할 수 있으나, 본 발명의 범위가 이에 한정되는 것은 아니다.

[0089] 통신부(320)는 RDMA 지원 네트워크(도 1의 130 참조)를 통하여 공유 메모리 서버(120)와 원격 직접 메모리 접근을 통한 분산 처리 장치들(도 1의 110 참조) 간의 송수신 신호를 전송하는데 필요한 통신 인터페이스를 제공한다.

[0090] 여기서, 통신부(320)는 다른 네트워크 장치와 유무선 연결을 통해 제어 신호 또는 데이터 신호와 같은 신호를 송수신하기 위해 필요한 하드웨어 및 소프트웨어를 포함하는 장치일 수 있다.

[0091] 이때, 통신부(320)는 RDMA 지원 네트워크(도 1의 130 참조)를 통해 원격 직접 메모리 접근을 통한 분산 처리 장치들(도 1의 110 참조)이 메모리(330)에 직접 접근하여 데이터를 읽고 쓸 수 있도록 지원할 수 있다.

[0092] 메모리(330)는 제어부(310)가 처리하는 데이터를 일시적 또는 영구적으로 저장하는 기능을 수행한다. 여기서, 메모리(330)는 자기 저장 매체(magnetic storage media) 또는 플래시 저장 매체(flash storage media)를 포함할 수 있으나, 본 발명의 범위가 이에 한정되는 것은 아니다.

[0093] 이때, 메모리(330)는 전체 또는 일부가 공유 메모리 버퍼 세그먼트로 할당되어, 다른 공유 메모리 서버들의 공유 메모리 버퍼 세그먼트들과 함께 공유 메모리 버퍼를 구성할 수 있다. 여기서, 공유 메모리 버퍼는 공유 메모리 버퍼 세그먼트들을 연결한 가상의 메모리 버퍼로, 실체는 각 공유 메모리 서버들(120)의 메모리(330)에 할당된 공유 메모리 버퍼 세그먼트들의 영역이다.

[0094] 이때, 메모리(330)에 할당된 공유 메모리 버퍼 세그먼트는 분산 처리 장치(도 1의 110 참조)의 로컬 공유 메모리 영역의 데이터가 변경됨에 따라 변경된 데이터가 동기화될 수 있다.

[0095] 공유 메모리 관리부(340)는 공유 메모리 버퍼를 구성하기 위하여 공유 메모리 버퍼 세그먼트를 할당하고 이를 관리한다.

[0096] 이때, 공유 메모리 관리부(340)는 분산 처리 장치(도 1의 110 참조)로부터 공유 메모리 버퍼의 생성 및 할당을 요청받은 경우, 주어진 공유 메모리 버퍼 세그먼트의 크기만큼 메모리(330)에서 공유 메모리 버퍼 세그먼트를 생성 및 할당하여 할당 정보를 반환할 수 있다.

[0097] 이때, 공유 메모리 관리부(340)는 공유 메모리 버퍼 세그먼트를 생성 요청에 대하여 공유 메모리 버퍼 생성키를 수신하고, 수신한 공유 메모리 버퍼 생성키가 이미 사용중이 아닌 경우에만 공유 메모리 버퍼 세그먼트를 생성 및 할당하여 할당 정보를 반환할 수 있다.

[0098] 이때, 공유 메모리 관리부(340)는 공유 메모리 버퍼 세그먼트의 접근 요청에 대하여 공유 메모리 버퍼 생성키를 수신하고, 수신한 공유 메모리 버퍼 생성키가 접근을 요청하는 공유 메모리 버퍼 세그먼트의 정보와 일치하는 경우에 해당 공유 메모리 버퍼 세그먼트의 접근을 허용할 수 있다. 예컨대, 공유 메모리 버퍼 생성키는 공유 메모리 버퍼 세그먼트의 크기를 의미할 수 있다.

[0099] 이때, 공유 메모리 관리부(340)는 공유 메모리 버퍼 세그먼트들 사이의 데이터 누적 연산을 수행할 수 있다. 예컨대, 제1 공유 메모리 버퍼 세그먼트로부터 제2 공유 메모리 버퍼 세그먼트로의 데이터 누적은, 제2 공유 메모리 버퍼 세그먼트를 잠그고 제1 공유 메모리 버퍼 세그먼트의 데이터를 제2 공유 메모리 버퍼 세그먼트에 누적함으로써 수행될 수 있다. 그리고, 데이터 누적 연산이 완료된 경우 연산 완료를 알리는 결과를 반환할 수 있다.

[0101] 도 4는 본 발명의 일 실시예에 따른 공유 메모리 버퍼를 구성하는 방법을 나타낸 도면이다.

[0102] 도 4를 참조하면, 본 발명의 일 실시예에 따른 공유 메모리 버퍼를 구성하는 방법은, 공유 메모리 버퍼의 생성 및 할당을 주도하는 마스터 분산 처리 장치(410)가 공유 메모리 서버들(440, 450, 460 및 470)에 공유 메모리 버퍼를 구성하기 위한 공유 메모리 버퍼 세그먼트들의 생성 및 할당을 요청한다.

[0103] 이때, 공유 메모리 서버 1(440)은 공유 메모리 버퍼 세그먼트 1(441)을 생성 및 할당하여 그 정보를 반환하고, 공유 메모리 서버 2(450)는 공유 메모리 버퍼 세그먼트 2(451)를 생성 및 할당하여 그 정보를 반환하고, 공유 메모리 서버 3(460)은 공유 메모리 버퍼 세그먼트 3(461)을 생성 및 할당하여 그 정보를 반환하고, 공유 메모리 서버 4(470)는 공유 메모리 버퍼 세그먼트 4(471)를 생성 및 할당하여 그 정보를 반환할 수 있다.

[0104] 이때, 각 공유 메모리 서버들(440, 450, 460 및 470)에서 공유 메모리 버퍼 세그먼트들(441, 451, 461 및 471)이 생성 및 할당되면, 이들은 연결되어 가상의 공유 메모리 버퍼(430)를 구성할 수 있다. 즉, 공유 메모리 버퍼(430)의 실체는 각 공유 메모리 서버들(440, 450, 460 및 470)에 할당된 공유 메모리 버퍼 세그먼트들(441, 451, 461 및 471)이다. 따라서, 공유 메모리 버퍼(430)에 대한 데이터 입출력은 공유 메모리 버퍼 세그먼트들(441, 451, 461 및 471)에 대한 데이터 입출력을 의미한다.

[0105] 이때, 마스터 분산 처리 장치(410)와 동일한 분산 처리 프레임워크에 속하는 다른 분산 처리 장치들은 슬레이브 분산 처리 장치(420)로 분류되며, 슬레이브 분산 처리 장치(420)는 마스터 분산 처리 장치(410)에 의하여 구성된 공유 메모리 버퍼의 정보를 획득하여 동일한 공유 메모리 버퍼를 이용할 수 있다.

[0106] 이때, 마스터 분산 처리 장치(410)와 슬레이브 분산 처리 장치(420)는 각각 자신의 메모리에 대하여 공유 메모리 버퍼와 동일한 크기의 로컬 공유 메모리 영역(411 및 421)을 할당할 수 있다. 그리고, 로컬 공유 메모리 영역(411 및 421)에 대하여 입출력을 수행하여 분산 처리 프레임워크에서의 분산 처리를 수행할 수 있다.

[0107] 이때, 각각의 로컬 공유 메모리 영역(411 및 421)은 공유 메모리 버퍼(430)와 동기화되어 유지되고, 분산 처리 장치들(410 및 420) 사이에서 공유 메모리 버퍼(430)를 통해 분산 처리 데이터를 공유할 수 있다. 특히, 분산 처리 장치들(410 및 420)은 RDMA를 통하여 공유 메모리 버퍼(430)에 대하여 직접 입출력하여 로컬 공유 메모리 영역(411 및 421)과 공유 메모리 버퍼(430) 사이의 동기화를 수행할 수 있다.

[0109] 도 5는 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 방법을 나타낸 동작 흐름도이다.

[0110] 도 5를 참조하면, 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 방법은 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버 클러스터를 등록한다(S501). 공유 메모리 서버 클러스터는 복수개의 공유 메모리 서버들(도 1의 120 참조)로 구성될 수 있다.

[0111] 또한, 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 방법은 원격 직접 메모리 접근을

통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 버퍼를 생성 및 할당한다(S503).

[0112] 또한, 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 방법은 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 주어진 분산 연산을 수행하고 공유 메모리 버퍼에 대하여 RDMA를 통한 직접 데이터 읽기 및 쓰기를 통하여 분산 처리 데이터를 공유한다(S505).

[0113] 또한, 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 방법은 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 버퍼의 사용을 종료하는 경우에 공유 메모리 버퍼를 해제 및 삭제한다(S507).

[0114] 또한, 본 발명의 일 실시예에 따른 원격 직접 메모리 접근을 통한 분산 처리 방법은 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버 클러스터의 등록을 해제한다(S509).

[0116] 도 6은 도 5에 도시된 공유 메모리 버퍼를 생성 및 할당하는 단계(S503)의 일 예를 나타낸 동작 흐름도이다.

[0117] 도 6을 참조하면, 도 5에 도시된 공유 메모리 버퍼를 생성 및 할당하는 단계(S503)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버별 공유 메모리 버퍼 세그먼트 크기를 계산한다(S601).

[0118] 또한, 도 5에 도시된 공유 메모리 버퍼를 생성 및 할당하는 단계(S503)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버들(도 1의 120 참조)에 공유 메모리 버퍼 세그먼트의 생성 및 할당을 요청한다(S603). 여기서, 공유 메모리 버퍼 세그먼트의 생성 및 할당 요청은 공유 메모리 버퍼의 생성 및 할당 요청과 동일한 의미로 사용될 수 있다.

[0119] 또한, 도 5에 도시된 공유 메모리 버퍼를 생성 및 할당하는 단계(S503)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버들(도 1의 120 참조)에 의하여 공유 메모리 버퍼 세그먼트들이 생성 및 할당되면 반환되는 정보를 획득한다(S605).

[0120] 또한, 도 5에 도시된 공유 메모리 버퍼를 생성 및 할당하는 단계(S503)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 모든 공유 메모리 버퍼 세그먼트들이 생성 및 할당되었는지 여부를 확인한다(S607).

[0121] 또한, 도 5에 도시된 공유 메모리 버퍼를 생성 및 할당하는 단계(S503)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 모든 공유 메모리 버퍼 세그먼트들이 생성 및 할당되어 공유 메모리 버퍼가 구성되면, 메모리에 공유 메모리 버퍼와 동일한 크기만큼의 로컬 공유 메모리 영역을 할당하고, 공유 메모리 버퍼와 로컬 공유 메모리 영역 사이의 메모리 맵핑 테이블을 갱신한다(S609).

[0123] 도 7은 도 5에 도시된 공유 메모리 버퍼를 해제 및 삭제하는 단계(S507)의 일 예를 나타낸 동작 흐름도이다.

[0124] 도 7을 참조하면, 도 5에 도시된 공유 메모리 버퍼를 해제 및 삭제하는 단계(S507)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버들(도 1의 120 참조)에 공유 메모리 버퍼 세그먼트의 해제 및 삭제를 요청한다(S701). 여기서, 공유 메모리 버퍼 세그먼트의 해제 및 삭제 요청은 공유 메모리 버퍼의 해제 및 삭제 요청과 동일한 의미로 사용될 수 있다.

[0125] 또한, 도 5에 도시된 공유 메모리 버퍼를 해제 및 삭제하는 단계(S507)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버들(도 1의 120 참조)에 의하여 공유 메모리 버퍼 세그먼트들이 해제 및 삭제되면 반환되는 정보를 획득한다(S703).

[0126] 또한, 도 5에 도시된 공유 메모리 버퍼를 해제 및 삭제하는 단계(S507)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 모든 공유 메모리 버퍼 세그먼트들이 해제 및 삭제되었는지 여부를 확인한다(S705).

[0127] 또한, 도 5에 도시된 공유 메모리 버퍼를 해제 및 삭제하는 단계(S507)는 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 모든 공유 메모리 버퍼 세그먼트들이 해제 및 삭제되어 공유 메모리 버퍼의 사용이 종료되면, 할당된 로컬 공유 메모리 영역을 해제하고, 상응하는 메모리 맵핑 테이블을 삭제한다(S707).

[0129] 도 8은 본 발명의 일 실시예에 따른 공유 메모리 버퍼들의 데이터 누적 연산 방법을 나타낸 동작이다.

[0130] 도 8을 참조하면, 본 발명의 일 실시예에 따른 공유 메모리 버퍼들의 데이터 누적 연산 방법은, 각 공유 메모리 서버들(810, 820 및 830)에서 할당된 공유 메모리 버퍼 세그먼트들에 대하여 데이터 누적 연산을 수행하는 것으

로 이루어진다.

[0131] 첫 번째 공유 메모리 서버(810)에는 제1 공유 메모리 버퍼 세그먼트 1(841) 및 제2 공유 메모리 버퍼 세그먼트 1(851)이 할당되어 있으며, 두 번째 공유 메모리 서버(820)에는 제1 공유 메모리 버퍼 세그먼트 2(842) 및 제2 공유 메모리 버퍼 세그먼트 2(852)가 할당되어 있으며, n 번째 공유 메모리 서버(830)에는 제1 공유 메모리 버퍼 세그먼트 n(843) 및 제2 공유 메모리 버퍼 세그먼트 n(853)이 할당되어 있다. 그리고, 제1 공유 메모리 버퍼 세그먼트 1(841), 제1 공유 메모리 버퍼 세그먼트 2(842) 및 제1 공유 메모리 버퍼 세그먼트 n(843) 등은 제1 공유 메모리 버퍼(840)를 구성한다. 또한, 제2 공유 메모리 버퍼 세그먼트 1(851), 제2 공유 메모리 버퍼 세그먼트 2(852) 및 제2 공유 메모리 버퍼 세그먼트 n(853) 등은 제2 공유 메모리 버퍼(850)를 구성한다.

[0132] 각 공유 메모리 서버들(810, 820 및 830)은 제1 공유 메모리 버퍼(840)의 제1 공유 메모리 버퍼 세그먼트(841, 842 및 843)의 데이터를 제2 공유 메모리 버퍼(850)의 제2 공유 메모리 버퍼 세그먼트(851, 852 및 853)에 누적하여 공유 메모리 버퍼들 간의 데이터 누적 연산을 수행할 수 있다.

[0134] 도 9는 도 8에 도시된 공유 메모리 버퍼들의 데이터 누적 연산 방법을 나타낸 동작 흐름도이다.

[0135] 도 9를 참조하면, 도 8에 도시된 공유 메모리 버퍼들의 데이터 누적 연산 방법은 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 제1 공유 메모리 버퍼의 데이터를 로컬 공유 메모리 영역과 동기화한다(S901).

[0136] 또한, 도 8에 도시된 공유 메모리 버퍼들의 데이터 누적 연산 방법은 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버들(도 1의 120 참조)에 제1 공유 메모리 버퍼로부터 제2 공유 메모리 버퍼로의 데이터 누적 연산을 요청한다(S903).

[0137] 또한, 도 8에 도시된 공유 메모리 버퍼들의 데이터 누적 연산 방법은 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 공유 메모리 서버들(도 1의 120 참조)로부터 제2 공유 메모리 버퍼 세그먼트를 잠근 이후 제1 공유 메모리 버퍼 세그먼트의 데이터를 제2 공유 메모리 버퍼 세그먼트에 누적한 결과를 수신한다(S905).

[0138] 또한, 도 8에 도시된 공유 메모리 버퍼들의 데이터 누적 연산 방법은 원격 직접 메모리 접근을 통한 분산 처리 장치(도 1의 110 참조)가, 모든 공유 메모리 버퍼 세그먼트들에 대한 누적 연산이 완료되었는지 확인하여 데이터 누적 연산의 결과를 반환한다(S907).

[0140] 본 발명에서 설명하는 특정 실행들은 실시예들로서, 어떠한 방법으로도 본 발명의 범위를 한정하는 것은 아니다. 명세서의 간결함을 위하여, 종래 전자적인 구성들, 제어시스템들, 소프트웨어, 상기 시스템들의 다른 기능적인 측면들의 기재는 생략될 수 있다. 또한, 도면에 도시된 구성 요소들 간의 선들의 연결 또는 연결 부재들은 기능적인 연결 및/또는 물리적 또는 회로적 연결들을 예시적으로 나타낸 것으로서, 실제 장치에서는 대체 가능하거나 추가의 다양한 기능적인 연결, 물리적인 연결, 또는 회로 연결들로서 나타내어질 수 있다. 또한, “필수적인”, “중요하게” 등과 같이 구체적인 언급이 없다면 본 발명의 적용을 위하여 반드시 필요한 구성 요소가 아닐 수 있다.

[0141] 따라서, 본 발명의 사상은 상기 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 또는 이로부터 등가적으로 변경된 모든 범위는 본 발명의 사상의 범주에 속한다고 할 것이다.

부호의 설명

[0142] 100: 원격 직접 메모리 접근을 통한 분산 처리 시스템

110: 원격 직접 메모리 접근을 통한 분산 처리 장치

120: 공유 메모리 서버 130: RDMA 지원 네트워크

210: 제어부 220: 통신부

230: 메모리 240: 연산 처리부

250: 공유 메모리 서버 접근 관리부

260: 메모리 맵핑 테이블 관리부

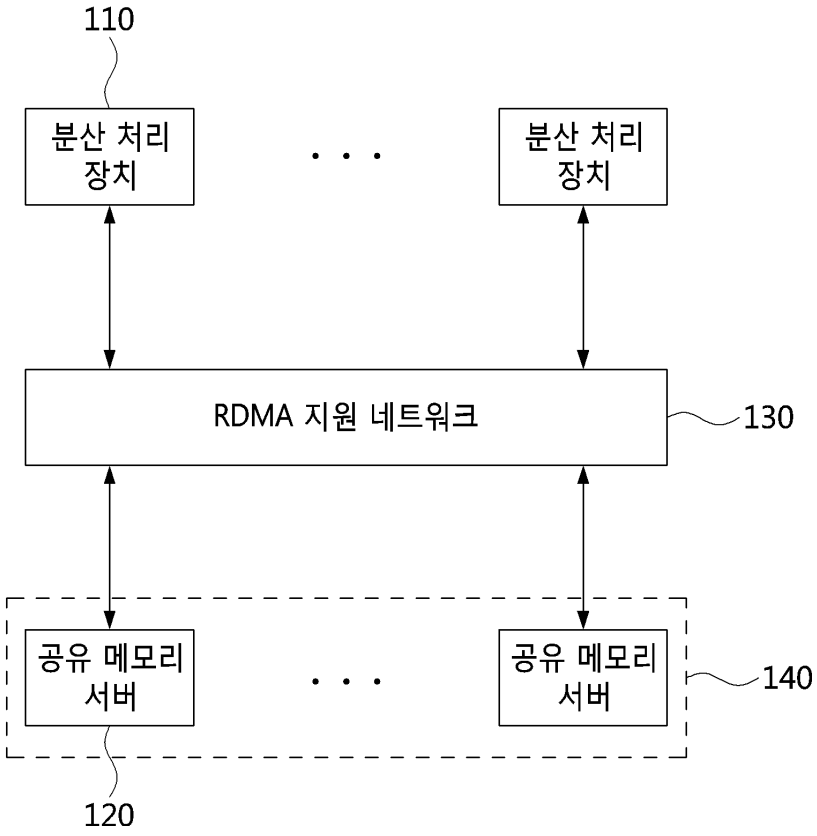
310: 제어부 320: 통신부

330: 메모리 340: 공유 메모리 관리부

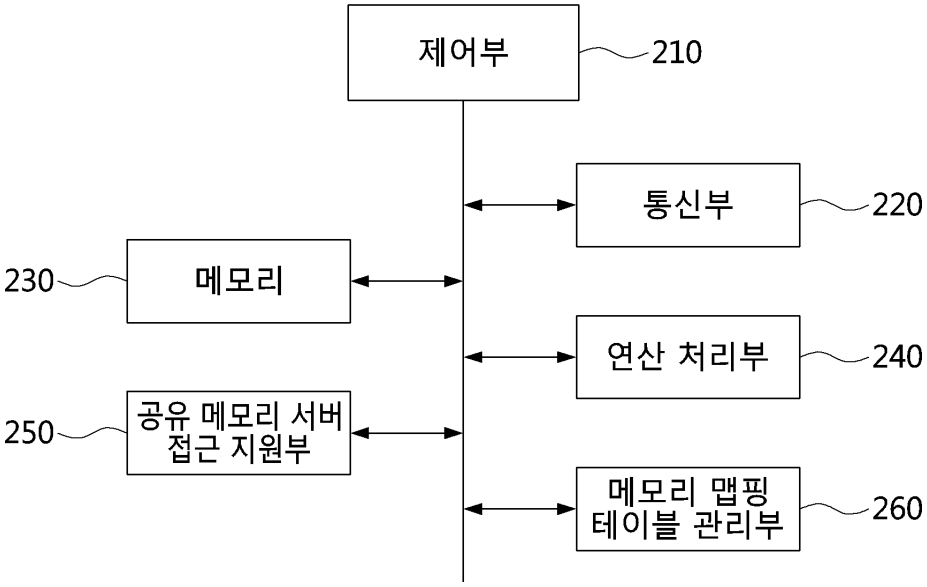
도면

도면1

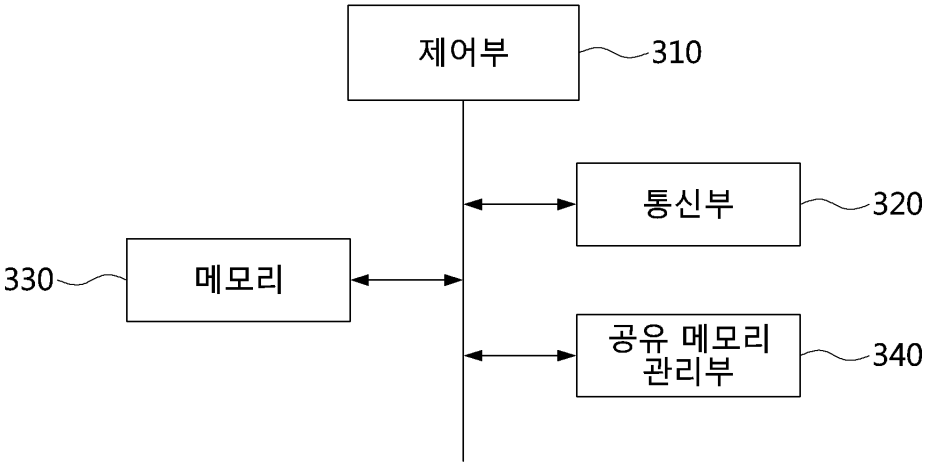
100



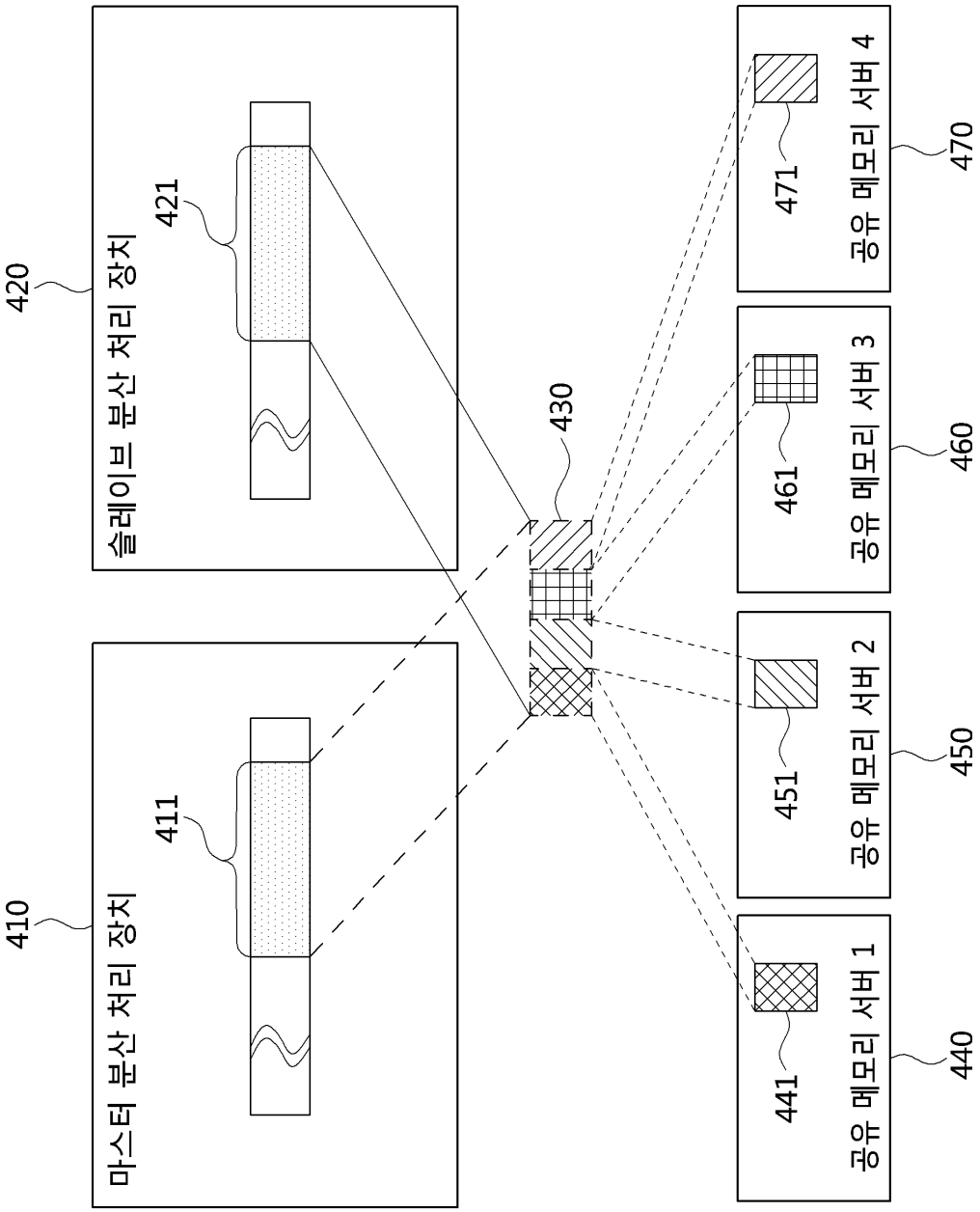
도면2


110
제어부
210
통신부
220
230
메모리
연산 처리부
240
250
공유 메모리 서버
접근 지원부
메모리 맵핑
테이블 관리부
260

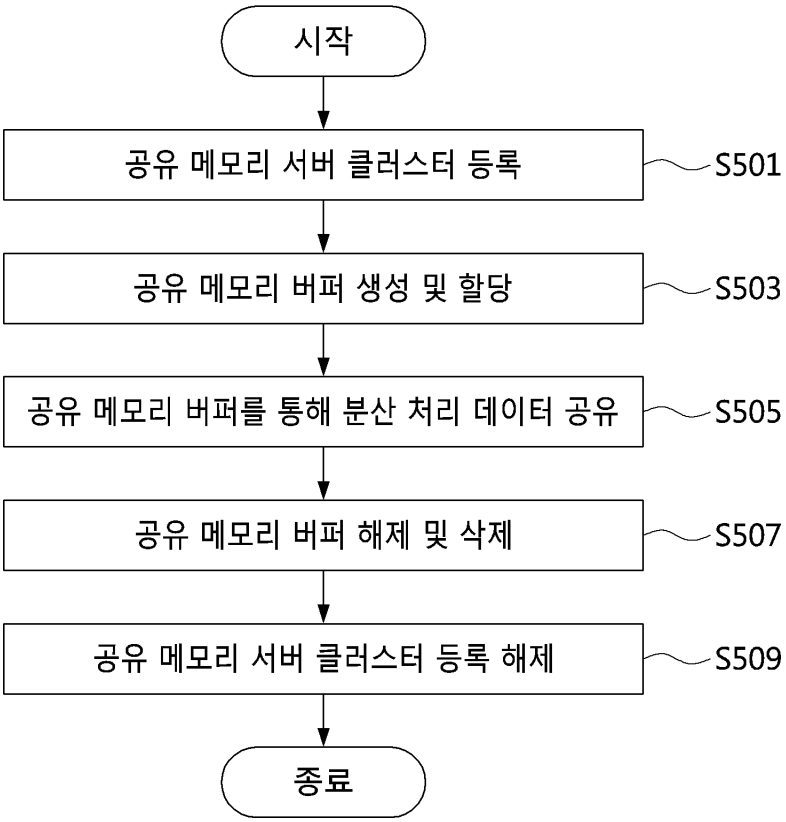
도면3


120
제어부
310
통신부
320
330
메모리
공유 메모리
관리부
340

도면4

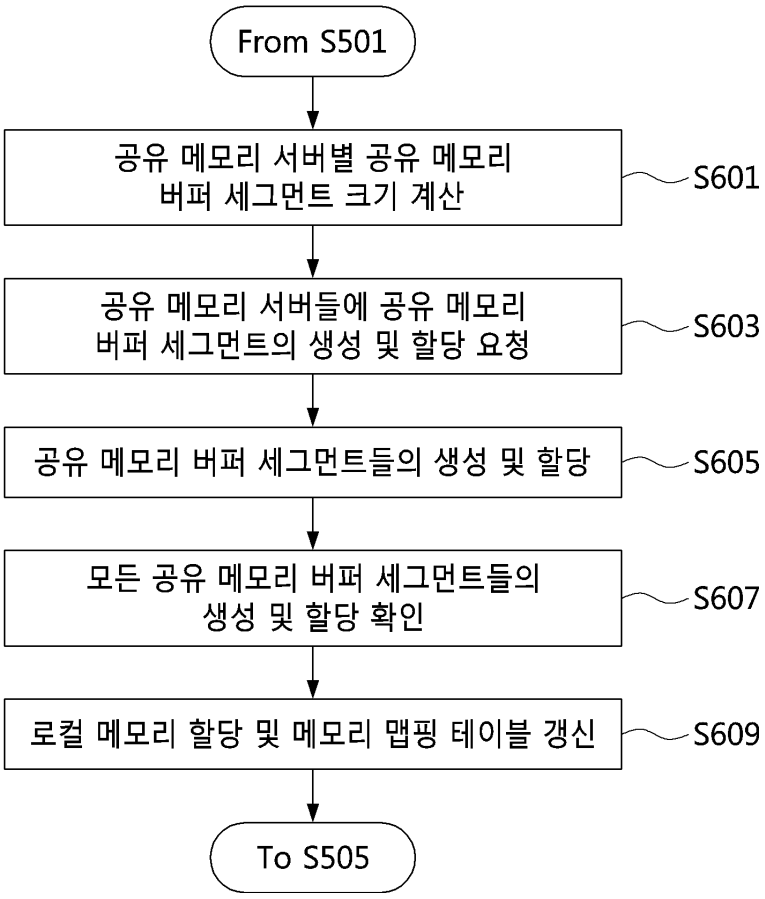

420
슬레이브 분산 처리 장치
421
410
마스터 분산 처리 장치
411
430
470
공유 메모리 서버 4
471
460
공유 메모리 서버 3
461
450
공유 메모리 서버 2
451
440
공유 메모리 서버 1
441

도면5


시작
공유 메모리 서버 클러스터 등록
S501
공유 메모리 버퍼 생성 및 할당
S503
공유 메모리 버퍼를 통해 분산 처리 데이터 공유
S505
공유 메모리 버퍼 해제 및 삭제
S507
공유 메모리 서버 클러스터 등록 해제
S509
종료

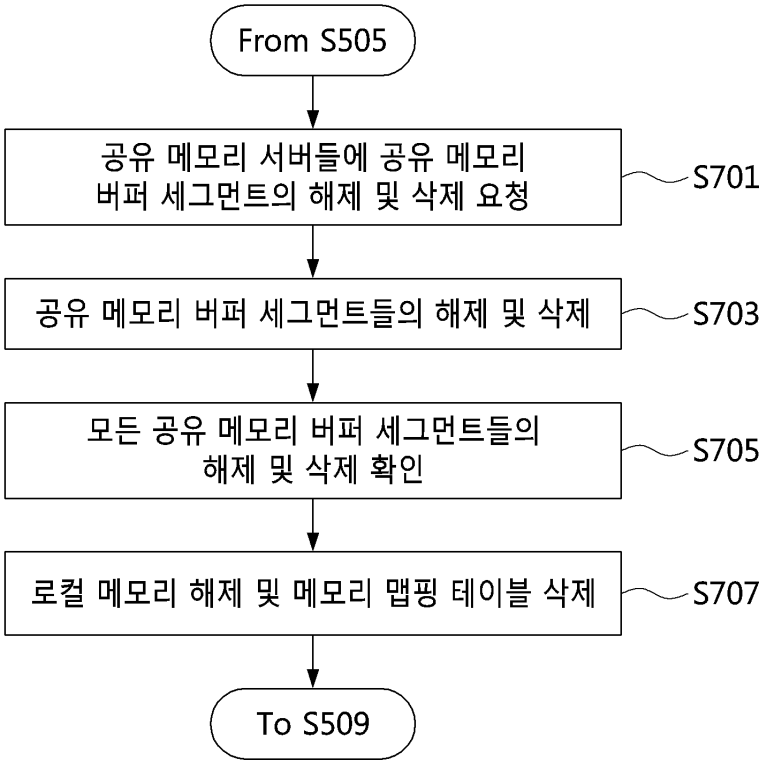
도면6

S503

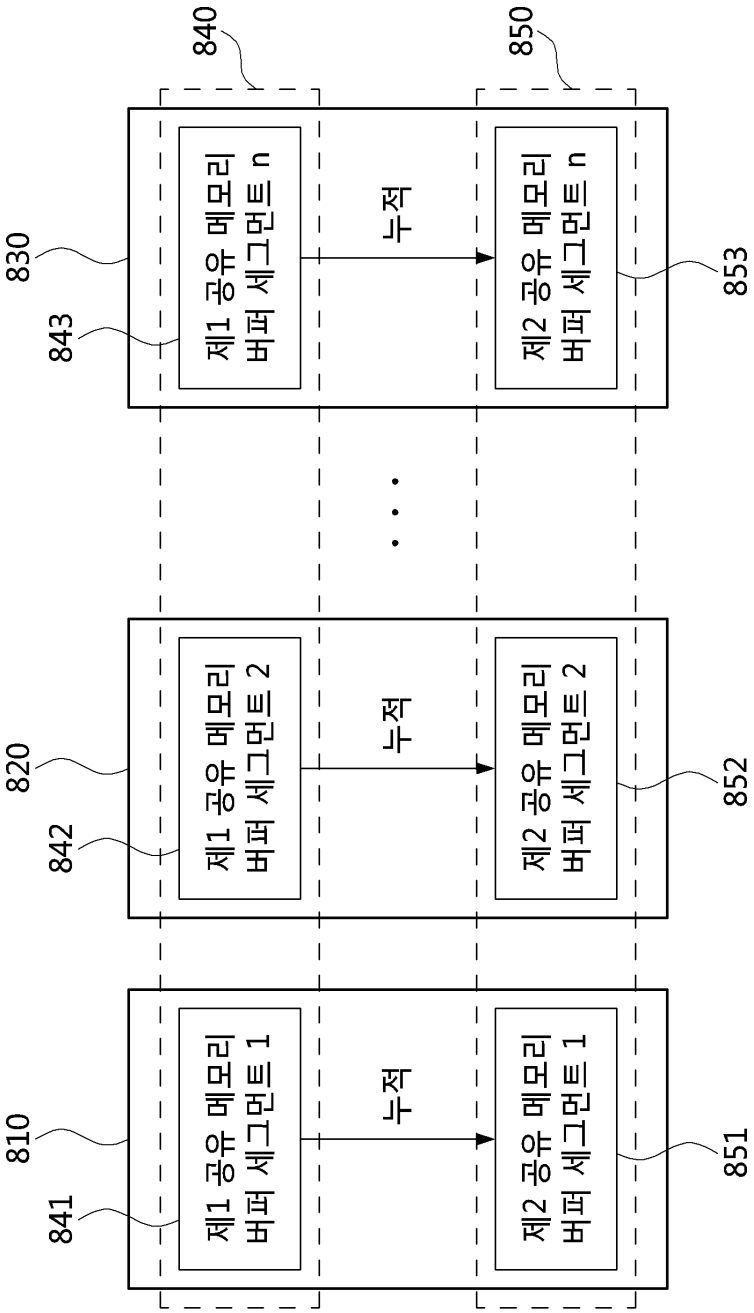

From S501
공유 메모리 서버별 공유 메모리 버퍼 세그먼트 크기 계산
S601
공유 메모리 서버들에 공유 메모리 버퍼 세그먼트의 생성 및 할당 요청
S603
공유 메모리 버퍼 세그먼트들의 생성 및 할당
S605
모든 공유 메모리 버퍼 세그먼트들의 생성 및 할당 확인
S607
로컬 메모리 할당 및 메모리 맵핑 테이블 갱신
S609
To S505

도면7

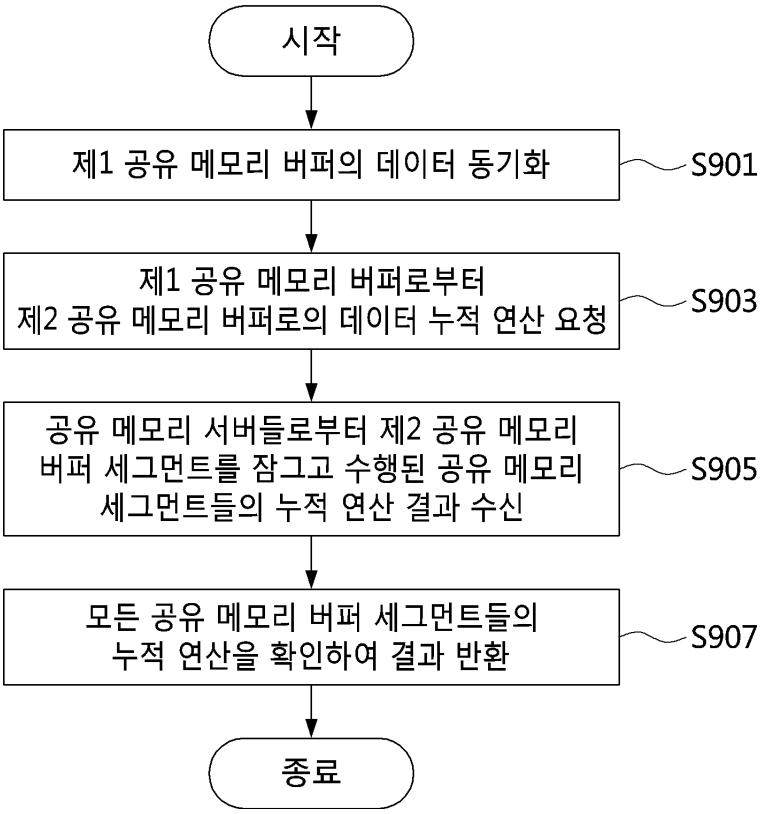
S507


From S505
공유 메모리 서버들에 공유 메모리 버퍼 세그먼트의 해제 및 삭제 요청
S701
공유 메모리 버퍼 세그먼트들의 해제 및 삭제
S703
모든 공유 메모리 버퍼 세그먼트들의 해제 및 삭제 확인
S705
로컬 메모리 해제 및 메모리 맵핑 테이블 삭제
S707
To S509

도면8


840
850
830
843
제1 공유 메모리
버퍼 세그먼트 n
누적
제2 공유 메모리
버퍼 세그먼트 n
853
820
842
제1 공유 메모리
버퍼 세그먼트 2
누적
제2 공유 메모리
버퍼 세그먼트 2
852
810
841
제1 공유 메모리
버퍼 세그먼트 1
누적
제2 공유 메모리
버퍼 세그먼트 1
851

도면9



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 10

【변경전】

청구항 9에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치로부터 상기 공유 메모리 버퍼를 생성하기 위한 공유 메모리 버퍼 세그먼트의 크기 정보와 함께 상기 공유 메모리 버퍼 세그먼트의 생성 및 할당을 요청을 수신하고, 상기 공유 메모리 버퍼 세그먼트를 생성 및 할당하여 상기 공유 메모리 버퍼를 구성하는 것을 특징으로 하는, 공유 메모리 서버.

【변경후】

청구항 8에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치로부터 상기 공유 메모리 버퍼를 생성하기 위한 공유 메모리 버퍼 세그먼트의 크기 정보와 함께 상기 공유 메모리 버퍼 세그먼트의 생성 및 할당을 요청을 수신하고, 상기 공유 메모리 버퍼 세그먼트를 생성 및 할당하여 상기 공유 메모리 버퍼를 구성하는 것을 특징으로 하는, 공유 메모리 서버.

【직권보정 2】

【보정항목】 청구범위

【보정세부항목】 청구항 11

【변경전】

청구항 9에 있어서,

상기 공유 메모리 버퍼는

연산에 의하여 특정 로컬 공유 메모리 영역의 데이터가 변경된 경우에 상기 변경된 로컬 공유 메모리 영역

의 데이터와 동기화되고, 변경된 데이터로 나머지 로컬 공유 메모리 영역들과 동기화되는 것을 특징으로 하는, 공유 메모리 서버.

【변경후】

청구항 8에 있어서,

상기 공유 메모리 버퍼는

연산에 의하여 특정 로컬 공유 메모리 영역의 데이터가 변경된 경우에 상기 변경된 로컬 공유 메모리 영역의 데이터와 동기화되고, 변경된 데이터로 나머지 로컬 공유 메모리 영역들과 동기화되는 것을 특징으로 하는, 공유 메모리 서버.

【직권보정 3】

【보정항목】 청구범위

【보정세부항목】 청구항 12

【변경전】

청구항 9에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치로부터 두 개 이상의 공유 메모리 버퍼들 사이의 데이터 누적 연산 요청을 수신하고, 상기 데이터 누적 연산의 대상이 되는 공유 메모리 버퍼 세그먼트들에 대하여 누적 연산을 수행하고, 결과를 상기 분산 처리 장치에 반환하는 것을 특징으로 하는, 공유 메모리 서버.

【변경후】

청구항 8에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치로부터 두 개 이상의 공유 메모리 버퍼들 사이의 데이터 누적 연산 요청을 수신하고, 상기 데이터 누적 연산의 대상이 되는 공유 메모리 버퍼 세그먼트들에 대하여 누적 연산을 수행하고, 결과를 상기 분산 처리 장치에 반환하는 것을 특징으로 하는, 공유 메모리 서버.

【직권보정 4】

【보정항목】 청구범위

【보정세부항목】 청구항 13

【변경전】

청구항 9에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치가 상기 공유 메모리 버퍼의 사용을 종료하기 위하여 전송한 상기 공유 메모리 버퍼 세그먼트의 해제 및 삭제 요청을 수신하여 상기 공유 메모리 버퍼 세그먼트를 해제 및 삭제하고, 결과를 상기 분산 처리 장치에 반환하여 상기 분산 처리 장치가 상기 로컬 공유 메모리 영역을 해제 및 삭제하고 상기 메모리 맵핑 테이블을 삭제하도록 하는 것을 특징으로 하는, 공유 메모리 서버.

【변경후】

청구항 8에 있어서,

상기 공유 메모리 관리부는

상기 분산 처리 장치가 상기 공유 메모리 버퍼의 사용을 종료하기 위하여 전송한 상기 공유 메모리 버퍼 세그먼트의 해제 및 삭제 요청을 수신하여 상기 공유 메모리 버퍼 세그먼트를 해제 및 삭제하고, 결과를 상기 분산 처리 장치에 반환하여 상기 분산 처리 장치가 상기 로컬 공유 메모리 영역을 해제 및 삭제하고 상기 메모리 맵핑 테이블을 삭제하도록 하는 것을 특징으로 하는, 공유 메모리 서버.

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년12월14일
(11) 등록번호 10-2190511
(24) 등록일자 2020년12월07일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2006.01) *G06F 9/50* (2018.01)
G06N 3/04 (2006.01)
(52) CPC특허분류
G06N 3/08 (2013.01)
G06F 9/5027 (2013.01)
(21) 출원번호 10-2019-0025730
(22) 출원일자 2019년03월06일
심사청구일자 2019년03월19일
(65) 공개번호 10-2020-0107124
(43) 공개일자 2020년09월16일
(56) 선행기술조사문헌
EP03392825 A2
(뒷면에 계속)
전체 청구항 수 : 총 9 항

(73) 특허권자
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
(72) 발명자
안신영
대전광역시 서구 둔산북로 160, 5동 701호
박유미
대전광역시 유성구 노은서로250번길 17-3
(뒷면에 계속)
(74) 대리인
한양특허법인

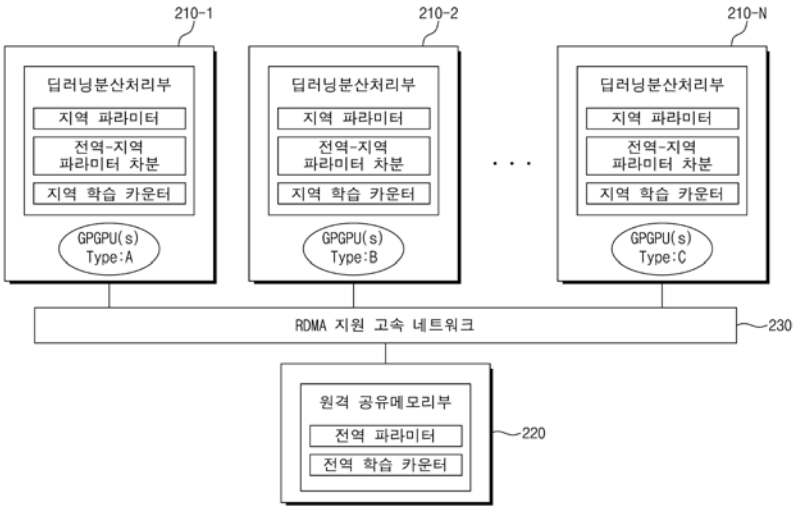
심사관 : 박성수

(54) 발명의 명칭 **이종 클러스터 기반의 분산 딥러닝 방법 및 이를 위한 장치**

(57) 요 약

이종 클러스터 기반의 분산 딥러닝 방법 및 이를 위한 장치가 개시된다. 본 발명의 일실시예에 따른 분산 딥러닝 방법은 딥러닝 성능이 상이한 복수개의 이기종 딥러닝 모듈들이 원격 공유 메모리를 기반으로 전역 파라미터와 전역 학습 카운터를 공유하는 단계; 상기 복수개의 이기종 딥러닝 모듈들이 상기 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하는 분산 딥러닝 학습과 원격 공유 메모리 업데이트를 중첩하여 수행하는 단계; 및 상기 원격 공유 메모리 업데이트에 의해 업데이트된 전역 학습 카운터를 고려하여 분산 딥러닝 프로세스를 종료하는 단계를 포함한다.

대 표 도



(52) CPC특허분류
G06N 3/0454 (2013.01)
(72) 발명자
임은지
대전광역시 유성구 노은동로 187, 602동 1801호
최용석
대전광역시 유성구 지족북로 60, 207동 303호

(56) 선행기술조사문헌
KR1020180131836 A
KR1020190087783 A
McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629, 2016.
Zheng, S et al. Asynchronous stochastic gradient descent with delay compensation. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 4120-4129. JMLR. org, 2017.

이 발명을 지원한 국가연구개발사업

과제고유번호	2016-0-00087
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원(IITP)
연구사업명	정보통신방송기술개발사업(SW컴퓨팅 산업원천기술개발사업)
연구과제명	대규모 딥러닝 고속 처리를 위한 HPC 시스템 개발
기 여 율	1/1
과제수행기관명	한국전자통신연구원
연구기간	2018.01.01 ~ 2018.12.31

명 세 서

청구범위

청구항 1

복수개의 분산 딥러닝 장치들에 의해 각 단계가 수행되는 분산 딥러닝 방법에 있어서,

딥러닝 성능이 상이한 상기 복수개의 분산 딥러닝 장치들이, 원격 공유 메모리를 기반으로 전역 파라미터와 전역 학습 카운터를 공유하는 단계;

상기 복수개의 분산 딥러닝 장치들 각각이, 상기 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하도록 상기 복수개의 분산 딥러닝 장치들 각각에 기저장된 지역 파라미터와 상기 전역 파라미터의 차분 연산 결과를 이용하여 분산 딥러닝 학습을 수행하고, 상기 분산 딥러닝 학습이 수행되는 동안 상기 차분 연산 결과를 이용하여 상기 전역 파라미터를 업데이트하는 분산 딥러닝 프로세스를 수행하는 단계; 및

상기 복수개의 분산 딥러닝 장치들이, 상기 분산 딥러닝 학습의 수행 횟수에 상응하는 상기 지역 학습 카운터를 기반으로 상기 전역 학습 카운터를 업데이트하고, 상기 업데이트된 전역 학습 카운터가 기설정된 종료 카운터 이상인지 여부를 판단하여 상기 분산 딥러닝 프로세스를 종료하는 단계

를 포함하는 것을 특징으로 하는 분산 딥러닝 방법.

청구항 2

청구항 1에 있어서,

상기 수행하는 단계는

상기 복수개의 분산 딥러닝 장치들 각각이, 차분 연산 결과를 이용하여 상기 지역 파라미터를 업데이트 하고, 상기 업데이트된 지역 파라미터를 이용하여 상기 분산 딥러닝 학습을 수행하고, 상기 분산 딥러닝 학습을 수행한 결과를 이용하여 상기 업데이트된 지역 파라미터를 재업데이트하는 것을 특징으로 하는 분산 딥러닝 방법.

청구항 3

청구항 1에 있어서,

상기 분산 딥러닝 방법은

상기 복수개의 분산 딥러닝 장치들에 각각 상기 지역 파라미터, 상기 차분 연산 결과 및 상기 지역 학습 카운터의 영역을 생성하는 단계;

상기 원격 공유 메모리에 전역 파라미터 영역 및 전역 학습 카운터 영역을 생성하는 단계; 및

상기 복수개의 분산 딥러닝 장치들 중 어느 하나가 상기 전역 파라미터 및 상기 전역 학습 카운터를 초기화하는 단계를 더 포함하는 것을 특징으로 하는 분산 딥러닝 방법.

청구항 4

청구항 1에 있어서,

상기 수행하는 단계는

상기 복수개의 분산 딥러닝 장치들이 각각 상기 분산 딥러닝 학습을 위한 딥러닝 학습 스레드(THREAD) 및 상기 원격 공유 메모리 업데이트를 위한 업데이트 스레드(THREAD)를 생성하는 단계를 포함하는 것을 특징으로 하는 분산 딥러닝 방법.

청구항 5

삭제

청구항 6

청구항 4에 있어서,

상기 수행하는 단계는

상기 딥러닝 학습 스레드 및 상기 업데이트 스레드 중 어느 하나로 할당되는 흐름제어락을 기반으로 하는 것을 특징으로 하는 분산 딥러닝 방법.

청구항 7

청구항 6에 있어서,

상기 흐름제어락은 상기 분산 딥러닝 프로세스가 시작된 이후에 상기 딥러닝 학습 스레드로 먼저 할당되는 것을 특징으로 하는 분산 딥러닝 방법.

청구항 8

청구항 1에 있어서,

상기 복수개의 분산 딥러닝 장치들은 원격 직접 메모리 접근(REMOTE DIRECT MEMORY ACCESS, RDMA)을 지원하는 고속 네트워크를 기반으로 상기 원격 공유 메모리에 접근하는 것을 특징으로 하는 분산 딥러닝 방법.

청구항 9

원격 공유 메모리의 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하도록 기저장된 지역 파라미터와 전역 파라미터의 차분 연산 결과를 이용하여 분산 딥러닝 학습을 수행하고,

상기 분산 딥러닝 학습이 수행되는 동안 상기 차분 연산 결과를 이용하여 상기 전역 파라미터를 업데이트하는 분산 딥러닝 프로세스를 수행하고,

상기 분산 딥러닝 학습의 수행 횟수에 상응하는 상기 지역 학습 카운터를 기반으로 상기 전역 학습 카운터를 업데이트하고, 상기 업데이트된 전역 학습 카운터가 기설정된 종료 카운터 이상인지 여부를 판단하여 상기 분산 딥러닝 프로세스를 종료하는 프로세서; 및

상기 지역 파라미터, 상기 전역 파라미터와 상기 지역 파라미터의 차분 연산 결과 및 상기 지역 학습 카운터를 저장하는 메모리

를 포함하는 것을 특징으로 하는 분산 딥러닝 장치.

청구항 10

청구항 9에 있어서,

상기 프로세서는

상기 차분 연산 결과를 이용하여 상기 지역 파라미터를 업데이트 하고, 상기 업데이트된 지역 파라미터를 이용하여 상기 분산 딥러닝 학습을 수행하고, 상기 분산 딥러닝 학습을 수행한 결과를 이용하여 상기 업데이트된 지역 파라미터를 재업데이트하는 것을 특징으로 하는 분산 딥러닝 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 분산 딥러닝 기술에 관한 것으로, 특히 이기종의 컴퓨팅 모듈로 구성되는 이종 HPC 클러스터 환경에서 효율적으로 분산 딥러닝을 수행할 수 있는 기술에 관한 것이다.

배 경 기 술

[0002] 딥러닝이란 사람의 신경세포(BIOLOGICAL NEURON)를 모사하여 기계가 학습하도록 하는 인공신경망(ARTIFICIAL NEURAL NETWORK) 기반의 기계 학습법이다. 최근 딥러닝 모델들은 응용의 인식 성능을 높이기 위해 대규모 모델로 진화하고 있으나 점차 대형화되는 딥러닝 모델과 대규모 학습 데이터를 단일 머신에서 처리하기에는 한계가 있다. 그래서 대규모 분산 컴퓨팅 자원을 활용하려는 노력의 일환으로 딥러닝 분산 플랫폼 기술이 개발되고 있

다.

[0003] 기존의 딥러닝 분산 처리는 대부분 동일한 규격과 성능의 클러스터를 가정하는 경우가 많다. 그러나 실제로 딥러닝 분산 처리를 하려고 할 때, 동일한 규격과 성능의 컴퓨팅 서버들로 구성된 클러스터를 구비하는 경우는 많지 않다. 따라서 이종 클러스터 환경에서 다른 규격의 서버들로 구성된 이종 클러스터를 동시에 모두 이용하여 효율적으로 딥러닝 분산처리를 수행하는 것은 쉬운 일이 아니다.

[0004] 일반적으로 동종 컴퓨터로 구성된 클러스터 환경에서는 동기식 파라미터 업데이트 방식을 이용한다. 그러나 동종 클러스터 환경에서 동시에 실행되는 분산 프로세스들도 시간이 지남에 따라 다양한 원인으로 인해 속도 차가 발생하기 때문에 동기식 트레이닝의 효율을 떨어뜨리게 된다. 이에 대한 대안으로 사용되는 것이 비동기식 파라미터 업데이트 방식이다. 비동기식 업데이트 방식은 파라미터 서버가 분산 컴퓨터들로부터 늦거나 빨리 도착하는 파라미터들의 동기를 맞추지 않고 트레이닝을 진행하는 방법이다. 비동기 방식은 동기식에 비해 정확성을 크게 희생시키지 않으면서 빠르게 트레이닝 할 수 있는 장점이 있다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 한국 등록 특허 제10-1559089호, 2015년 10월 2일 공개(명칭: 장치의 컴포넌트들 간에 메모리 자원들을 공유하기 위한 통신 프로토콜)

발명의 내용

해결하려는 과제

[0006] 본 발명의 목적은 이기종의 GPU를 이용한 분산 딥러닝 수행 시 통신 오버헤드를 감소시킬 수 있는 효과적인 분산 딥러닝 방법을 제공하는 것이다.

[0007] 또한, 본 발명의 목적은 동시에 성능이 다른 GPU들을 효과적으로 사용할 수 있는 분산 딥러닝 방법을 제공하는 것이다.

[0008] 또한, 본 발명의 목적은 학습 속도가 다른 분산 프로세스들이 전체 학습을 효과적으로 나누어 수행할 수 있도록 하는 것이다.

[0009] 또한, 본 발명의 목적은 학습한 파라미터의 업데이트를 지연하는 방식으로 계산과 통신을 중첩하여 각각의 GPU 활용률을 극대화함으로써 우수한 분산 처리 확장성을 제공하는 것이다.

과제의 해결 수단

[0010] 상기한 목적을 달성하기 위한 본 발명에 따른 분산 딥러닝 방법은 딥러닝 성능이 상이한 복수개의 이기종 딥러닝 모듈들이 원격 공유 메모리를 기반으로 전역 파라미터와 전역 학습 카운터를 공유하는 단계; 상기 복수개의 이기종 딥러닝 모듈들이 상기 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하는 분산 딥러닝 학습과 원격 공유 메모리 업데이트를 중첩하여 수행하는 단계; 및 상기 원격 공유 메모리 업데이트에 의해 업데이트된 전역 학습 카운터를 고려하여 분산 딥러닝 프로세스를 종료하는 단계를 포함한다.

[0011] 이 때, 분산 딥러닝 방법은 상기 복수개의 이기종 딥러닝 모듈들에 각각 지역 파라미터, 전역-지역 파라미터 차분 및 지역 학습 카운터 영역을 생성하는 단계; 및 상기 원격 공유 메모리에 전역 파라미터 영역 및 전역 학습 카운터 영역을 생성하는 단계를 더 포함할 수 있다.

[0012] 이 때, 분산 딥러닝 방법은 상기 복수개의 이기종 딥러닝 모듈들 중 어느 하나의 마스터 모듈을 통해 상기 전역 파라미터 및 상기 전역 학습 카운터를 초기화하는 단계를 더 포함할 수 있다.

[0013] 이 때, 수행하는 단계는 상기 복수개의 이기종 딥러닝 모듈들이 각각 상기 분산 딥러닝 학습을 위한 딥러닝 학습 스레드(THREAD) 및 상기 원격 공유 메모리 업데이트를 위한 업데이트 스레드(THREAD)를 생성하는 단계를 포함할 수 있다.

[0014] 이 때, 초기화하는 단계는 상기 업데이트 스레드의 웨이크업 시점을 기준으로 수행될 수 있다.

[0015] 이 때, 수행하는 단계는 상기 딥러닝 학습 스레드 및 상기 업데이트 스레드 중 어느 하나로 할당되는 흐름제어락을 기반으로 할 수 있다.

[0016] 이 때, 흐름제어락은 상기 분산 딥러닝 프로세스가 시작된 이후에 상기 딥러닝 학습 스레드로 먼저 할당될 수 있다.

[0017] 이 때, 복수개의 이기종 딥러닝 모듈들은 원격 직접 메모리 접근(REMOTE DIRECT MEMORY ACCESS, RDMA)을 지원하는 고속 네트워크를 기반으로 상기 원격 공유 메모리에 접근할 수 있다.

[0018] 또한, 본 발명의 일실시예에 따른 분산 딥러닝 장치는, 원격 공유 메모리의 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하는 분산 딥러닝 학습과 원격 공유 메모리 업데이트를 중첩하여 수행하고, 상기 원격 공유 메모리 업데이트를 기반으로 업데이트된 전역 학습 카운터를 고려하여 분산 딥러닝 프로세스를 종료하는 프로세서; 및 지역 파라미터, 전역-지역 파라미터 차분 및 지역 학습 카운터를 저장하는 메모리를 포함한다.

[0019] 이 때, 프로세서는 지역 파라미터, 전역-지역 파라미터 차분 및 지역 학습 카운터 영역을 생성하고, 상기 원격 공유 메모리에 전역 파라미터 영역 및 전역 학습 카운터 영역을 생성할 수 있다.

[0020] 이 때, 프로세서는 상기 전역 파라미터 및 상기 전역 학습 카운터를 초기화할 수 있다.

[0021] 이 때, 프로세서는 상기 분산 딥러닝 학습을 위한 딥러닝 학습 스레드(THREAD) 및 상기 원격 공유 메모리 업데이트를 위한 업데이트 스레드(THREAD)를 생성할 수 있다.

[0022] 이 때, 프로세서는 상기 업데이트 스레드의 웨이크업 시점을 기준으로 상기 초기화를 수행할 수 있다.

[0023] 이 때, 프로세서는 상기 딥러닝 학습 스레드 및 상기 업데이트 스레드 중 어느 하나로 흐름제어락을 할당하여 상기 분산 딥러닝 학습 및 상기 원격 공유 메모리 업데이트를 수행할 수 있다.

[0024] 이 때, 흐름제어락은 상기 분산 딥러닝 프로세스가 시작된 이후에 상기 딥러닝 학습 스레드로 먼저 할당될 수 있다.

[0025] 이 때, 프로세서는 원격 직접 메모리 접근(REMOTE DIRECT MEMORY ACCESS, RDMA)을 지원하는 고속 네트워크를 기반으로 상기 원격 공유 메모리에 접근할 수 있다.

발명의 효과

[0026] 본 발명에 따르면, 이기종의 GPU를 이용한 분산 딥러닝 수행 시 통신 오버헤드를 감소시킬 수 있는 효과적인 분산 딥러닝 방법을 제공할 수 있다.

[0027] 또한, 본 발명은 동시에 성능이 다른 GPU들을 효과적으로 사용할 수 있는 분산 딥러닝 방법을 제공할 수 있다.

[0028] 또한, 본 발명은 학습 속도가 다른 분산 프로세스들이 전체 학습을 효과적으로 나누어 수행할 수 있도록 할 수 있다.

[0029] 또한, 본 발명은 학습한 파라미터의 업데이트를 지연하는 방식으로 계산과 통신을 중첩하여 각각의 GPU 활용률을 극대화함으로써 우수한 분산 처리 확장성을 제공할 수 있다.

도면의 간단한 설명

[0030] 도 1은 본 발명의 일실시예에 따른 분산 딥러닝 방법을 나타낸 동작흐름도이다.

도 2는 본 발명의 일실시예에 따른 분산 딥러닝 시스템을 나타낸 도면이다.

도 3은 본 발명의 일실시예에 따른 분산 딥러닝 과정을 상세하게 나타낸 동작흐름도이다.

도 4는 본 발명에 따른 딥러닝 학습 스레드를 기반으로 분산 딥러닝을 수행하는 과정의 일 예를 상세하게 나타낸 동작흐름도이다.

5는 본 발명에 따른 업데이트 스레드를 기반으로 원격 공유 메모리를 업데이트하는 과정의 일 예를 상세하게 나타낸 동작흐름도이다.

도 6은 본 발명의 일실시예에 따른 분산 딥러닝 장치를 나타낸 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0031] 본 발명을 첨부된 도면을 참조하여 상세히 설명하면 다음과 같다. 여기서, 반복되는 설명, 본 발명의 요지를 불필요하게 흐릴 수 있는 공지 기능, 및 구성에 대한 상세한 설명은 생략한다. 본 발명의 실시형태는 당 업계에서 평균적인 지식을 가진 자에게 본 발명을 보다 완전하게 설명하기 위해서 제공되는 것이다. 따라서, 도면에서의 요소들의 형상 및 크기 등은 보다 명확한 설명을 위해 과장될 수 있다.

[0032] 이하, 본 발명에 따른 바람직한 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0034] 도 1은 본 발명의 일실시예에 따른 분산 딥러닝 방법을 나타낸 동작흐름도이다.

[0035] 도 1을 참조하면, 본 발명의 일실시예에 따른 분산 딥러닝 방법은 딥러닝 성능이 상이한 복수개의 이기종 딥러닝 모듈들이 원격 공유 메모리를 기반으로 전역 파라미터와 전역 학습 카운터를 공유한다(S110).

[0036] 이 때, 원격 공유 메모리에 저장된 전역 파라미터와 전역 학습 카운터는 배타적으로 업데이트가 가능한 데이터에 상응하는 것으로, 처리 성능이 서로 상이한 복수개의 이기종 딥러닝 모듈들이 전체 학습을 효과적으로 나누어 수행할 수 있도록 할 수 있다.

[0037] 이 때, 복수개의 이기종 딥러닝 모듈들은 원격 직접 메모리 접근(REMOTE DIRECT MEMORY ACCESS, RDMA)을 지원하는 고속 네트워크를 기반으로 원격 공유 메모리에 접근할 수 있다. 따라서, 원격 공유 메모리는 전역 파라미터와 전역 학습 카운터를 복수개의 이기종 딥러닝 모듈들에게 제공하여 직접 접근할 수 있도록 지원할 수 있다.

[0038] 예를 들어 도 2를 참조하면, 본 발명의 일실시예에 따른 복수개의 이기종 딥러닝 모듈들(210-1~210-N)은 RDMA 고속 네트워크(230)를 통해 원격 공유 메모리(220)에 접근할 수 있다. 이 때, 도 2에 도시된 것처럼 본 발명의 일실시예에 따른 복수개의 이기종 딥러닝 모듈들(210-1~210-N)은 딥러닝 학습을 수행하는 계산노드에 해당할 수 있으며, 상호간에 서로 다른 성능의 GPGPU(GENERAL PURPOSE COMPUTING ON GRAPHICS PROCESSING UNITS))들을 포함할 수 있다.

[0039] 이 때, 도 1에는 도시하지 아니하였으나, 본 발명의 일실시예에 따른 분산 딥러닝 방법은 복수개의 이기종 딥러닝 모듈들이 각각 지역 파라미터, 전역-지역 파라미터 차분 및 지역 학습 카운터 영역을 생성한다. 예를 들어, 도 2에 도시된 것처럼, 본 발명의 일실시예에 따른 복수개의 이기종 딥러닝 모듈들(210-1~210-N)은 각각 지역 파라미터, 전역-지역 파라미터 차분, 지역 학습 카운터를 포함할 수 있다.

[0040] 이 때, 학습 카운터란, 분산 딥러닝 프로세스들이 딥러닝 학습을 수행할 때 학습한 미니배치(MINI-BATCH)의 전체 횟수를 카운팅하는데 사용될 수 있으며, 각각의 분산 딥러닝 프로세스들이 학습해야 할 미니배치의 순서 번호를 할당 받는데 활용될 수 있다. 이와 같이 각각의 딥러닝 모듈로 할당된 학습 카운터는 지역 학습 카운터로써 저장될 수 있다. 이 때, 각각의 딥러닝 모듈에 포함된 분산 프로세스들이 수행해야 할 전체 미니배치의 횟수는 분산 딥러닝 프로세스를 이용하는 사용자가 지정할 수 있다.

[0041] 또한, 본 발명의 일실시예에 따른 분산 딥러닝 방법은 복수개의 이기종 딥러닝 모듈들이 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하는 분산 딥러닝 학습과 원격 공유 메모리 업데이트를 중첩하여 수행한다(S120).

[0042] 이 때, 복수개의 이기종 딥러닝 모듈들은 분산 딥러닝 학습을 통해 지역 파라미터를 자체적으로 학습시킬 수 있고, 원격 공유 메모리에 보관되는 전역 학습 카운터는 복수개의 이기종 딥러닝 모듈들 각각에 저장된 지역 학습 카운터와 비교하여 동일하면 변경하는 방식(COMPARE AND SWAP)으로 업데이트될 수 있다.

[0043] 일반적으로 분산 딥러닝 플랫폼에서 분산 딥러닝 학습을 수행하는 프로세스들은 상호간에 대규모 파라미터를 빈번하게 송수신해야 하므로 이 과정에서 발생하는 통신 오버헤드는 전체 분산 딥러닝 학습 성능에서 차지하는 비중이 매우 높은 형편이다. 따라서, 효과적인 분산 딥러닝 학습을 위해서는 통신시간을 감소시키거나 또는 통신시간과 계산시간을 중첩함으로써 통신시간을 숨길 필요가 있다.

[0044] 이와 같은 문제점을 해결하기 위해, 본 발명에서는 학습된 파라미터의 업데이트를 즉각적으로 수행하지 않고 지연하는 방식으로 계산과 통신을 중첩시키는 분산 딥러닝 방법을 제안하고자 한다.

[0045] 이 때, 복수개의 이기종 딥러닝 모듈들이 각각 분산 딥러닝 학습을 위한 딥러닝 학습 스레드(THREAD) 및 원격 공유 메모리 업데이트를 위한 업데이트 스레드(THREAD)를 생성할 수 있다. 일반적으로 복수개의 이기종 딥러닝 모듈들 각각의 메인 스레드가 분산 딥러닝 학습 스레드에 상응할 수 있다.

[0046] 또한, 분산 딥러닝 학습 및 원격 공유 메모리 업데이트는 딥러닝 학습 스레드 및 업데이트 스레드 중 어느 하나

로 할당되는 흐름제어락을 기반으로 수행될 수 있다.

[0047] 이하에서는 도 4 내지 도 5를 기반으로 분산 딥러닝 학습과 원격 공유 메모리 업데이트를 중첩 수행하는 두 개의 스레드들의 세부 절차를 설명하도록 한다.

[0048] 먼저, 도 4에 도시된 것처럼, 딥러닝 학습 스레드는 시작되면(S410) 먼저 흐름제어락을 획득할 수 있다(S420).

[0049] 이 때, 흐름제어락은 딥러닝 학습 스레드와 업데이트 스레드 간의 흐름제어를 위해 사용되는 것으로, 흐름제어락은 분산 딥러닝 프로세스가 시작된 이후에 딥러닝 학습 스레드로 먼저 할당될 수 있다.

[0050] 이 후, 전역-지역 파라미터 차분으로부터 지역 파라미터를 업데이트할 수 있다(S430). 이 후, 딥러닝 학습 스레드로 할당되었던 흐름제어락을 해제하고, 업데이트 스레드로 웨이크업 신호를 보내 깨워줄 수 있다(S440). 즉, 분산 딥러닝 학습 스레드는 분산 딥러닝 학습을 시작하기 전에 업데이트 스레드를 깨워준다.

[0051] 이 후, 업데이트된 지역 파라미터를 이용하여 하나의 미니배치 데이터에 대한 분산 딥러닝 학습 수행한 뒤(S450), 학습 결과를 기반으로 지역 파라미터 업데이트할 수 있다(S460).

[0052] 이 후, 원격 공유 메모리에 저장된 전역 학습 카운터 고려하여 추가적인 분산 딥러닝 프로세스가 필요한 경우에는 딥러닝 학습 스레드 반복 수행하되, 전역 학습 카운터가 만료되어 사용자가 지정한 미니배치에 도달하였을 경우에는 딥러닝 분산 프로세스를 종료할 수 있다(S470).

[0053] 또한, 도 5를 참조하면, 업데이트 스레드는 생성된 이후에 딥러닝 학습 스레드 또는 메인 스레드가 깨워줄 때까지 대기할 수 있다(S510).

[0054] 따라서, 웨이크업 신호가 발생하는지 여부를 판단하고(S515), 웨이크업 신호가 발생하여 업데이트 스레드가 깨어나면, 종료변수가 참인지 여부를 확인할 수 있다(S525).

[0055] 단계(S525)의 판단결과 종료변수가 참이면, 업데이트 스레드 종료할 수 있다(S570).

[0056] 또한, 단계(S525)의 판단결과 종료변수가 참이 아니면, 업데이트 스레드로 흐름제어락을 할당할 수 있다(S530).

[0057] 이 후, 원격 공유 메모리에 저장되어 있는 전역 파라미터를 딥러닝 모듈의 지역 버퍼로 읽어와서 전역 파라미터와 지역 파라미터의 차분을 계산할 수 있다(S540).

[0058] 이 후, 단계(S540)을 통해 산출된 전력-지역 파라미터 차분을 이용하여 원격 공유 메모리의 전역 파라미터가 증가하도록 업데이트한 뒤(S550) 업데이트 스레드로 할당된 흐름제어락을 해제할 수 있다(S560).

[0059] 이 때, 단계(S530) 내지 단계(S560)의 절차는 분산 딥러닝 학습이 완료될 때까지 반복적으로 수행될 수 있으며, 대체로 분산 딥러닝 학습시간이 원격 공유 메모리 업데이트 시간보다 길기 때문에 업데이트 스레드는 전역 파라미터 업데이트 완료 후에 대기상태로 회귀할 수 있다.

[0060] 이와 같이, 본 발명에서는 복수개의 이기종 딥러닝 모듈들 각각에서 N번째로 학습한 파라미터가 N+1번째 학습 도중에 전역 파라미터로 업데이트될 수 있고, N번째 학습 도중에 읽어온 전역 파라미터를 N+1번째 학습 전에 지역 파라미터로써 업데이트하여 학습을 수행할 수 있다. 따라서, 종래에 파라미터 서버를 이용하는 비동기 방식처럼 새로운 전역 파라미터가 업데이트될 때까지 대기할 필요가 없으므로 통신에 의해 지체되는 시간을 절약할 수 있다. 즉, 본 발명에서는 딥러닝 학습 스레드와 업데이트 스레드가 흐름제어락과 대기/웨이크업 방식을 이용하여 학습과 통신(전역 파라미터 업데이트)을 중첩 수행할 수 있다.

[0061] 또한, 본 발명의 일실시예에 따른 분산 딥러닝 방법은 원격 공유 메모리 업데이트에 의해 업데이트된 전역 학습 카운터를 고려하여 분산 딥러닝 프로세스를 종료한다(S130).

[0062] 예를 들어, 도 2에 도시된 것과 같은 본 발명의 딥러닝 모듈(210-1~210-N)은 원격 공유 메모리에 저장된 전역 학습 카운터를 배타적으로 증가시키면서 분산 딥러닝 학습을 수행하므로, 전역 학습 카운터가 사용자가 지정한 값에 도달하였을 때에 분산 딥러닝 프로세스를 종료할 수 있다.

[0063] 이와 같이 함으로써 저속 GPU는 전체 미니배치 횟수 중에서 더 적은 횟수를 학습하고, 고속 GPU는 더 많은 미니배치를 학습할 수 있으므로 분산 딥러닝 프로세스의 사용자가 지정한 미니배치에 도달하였을 때에 이기종의 분산 딥러닝 프로세스들은 거의 동시에 분산 딥러닝 학습을 종료할 수 있다.

[0064] 또한, 도 1에는 도시하지 아니하였으나, 본 발명의 일실시예에 따른 분산 딥러닝 방법은 원격 공유 메모리에 전역 파라미터 영역 및 전역 학습 카운터 영역을 생성한다.

[0065] 예를 들어, 도 2에 도시된 것과 같은 본 발명의 일실시예에 따른 복수개의 이기종 딥러닝 모듈들(210-1~210-N)은 RDMA 고속 네트워크(230)를 기반으로 원격 공유 메모리(220)로 접근하여 전역 파라미터 영역과 전역 학습 카운터 영역을 생성할 수 있다. 이 때, 복수개의 이기종 딥러닝 모듈들(210-1~210-N) 중 어느 하나의 마스터 모듈을 설정하고, 설정된 마스터 모듈을 이용하여 전역 파라미터 영역과 전역 학습 카운터 영역을 생성할 수도 있다.

[0066] 또한, 도 1에는 도시하지 아니하였으나, 본 발명의 일실시예에 따른 분산 딥러닝 방법은 복수개의 이기종 딥러닝 모듈들 중 어느 하나의 마스터 모듈을 통해 전역 파라미터 및 전역 학습 카운터를 초기화한다.

[0067] 이 때, 전역 파라미터는 딥러닝 모듈별 또는 딥러닝 플랫폼별로 다양한 방식으로 초기화될 수 있으며, 전역 학습 카운터는 0으로 초기화될 수 있다.

[0068] 또한, 이와 같은 초기화 과정은 업데이트 스레드가 최초로 웨이크업되는 시점을 기준으로 수행될 수도 있다.

[0069] 또한, 도 1에는 도시하지 아니하였으나, 본 발명의 일실시예에 따른 분산 딥러닝 방법은 상술한 바와 같이 본 발명의 실시예에 따른 분산 딥러닝 과정에서 발생되는 다양한 정보를 저장할 수 있다.

[0070] 이와 같은 이종 클러스터 기반의 분산 딥러닝 방법을 통해 이기종의 GPU를 이용한 분산 딥러닝 수행 시 통신 오버헤드를 감소시킬 수 있다.

[0071] 또한, 동시에 성능이 다른 GPU들을 효과적으로 사용할 수 있는 분산 딥러닝 방법을 제공할 수 있다.

[0072] 또한, 학습 속도가 다른 분산 프로세스들이 전체 학습을 효과적으로 나누어 수행할 수 있도록 할 수 있다.

[0073] 또한, 학습한 파라미터의 업데이트를 지연하는 방식으로 계산과 통신을 중첩하여 각각의 GPU 활용률을 극대화함으로써 우수한 분산 처리 확장성을 제공할 수 있다.

[0075] 도 3은 본 발명의 일실시예에 따른 분산 딥러닝 과정을 상세하게 나타낸 동작흐름도이다.

[0076] 도 3을 참조하면, 본 발명의 일실시예에 따른 분산 딥러닝 과정은 먼저 복수개의 이기종 딥러닝 모듈들에 각각 지역 파라미터, 전역-지역 파라미터 차분, 지역 학습 카운터 영역을 생성한다(S310).

[0077] 이 후, 복수개의 이기종 딥러닝 모듈들을 통해 원격 공유 메모리에 전역 파라미터 영역, 전역 학습 카운터 영역을 생성한다(S320).

[0078] 이 후, 복수개의 이기종 딥러닝 모듈들 중 어느 하나의 마스터 모듈을 통해 전역 파라미터와 전역 학습 카운터를 초기화한다(S330).

[0079] 이 때, 전역 파라미터는 딥러닝 모듈별 또는 딥러닝 플랫폼별로 다양한 방식으로 초기화될 수 있으며, 전역 학습 카운터는 0으로 초기화될 수 있다.

[0080] 이 후, 복수개의 이기종 딥러닝 모듈들은 각각 분산 딥러닝 학습을 위한 딥러닝 학습 스레드와 원격 공유 메모리 업데이트를 위한 업데이트 학습 스레드를 생성한다(S340).

[0081] 이 후, 복수개의 이기종 딥러닝 모듈들은 딥러닝 학습 스레드를 통해 분산 딥러닝 학습을 수행하기 이전에 웨이크업 신호를 발생시켜 업데이트 스레드를 깨운다(S350).

[0082] 이 후, 복수개의 이기종 딥러닝 모듈들은 각각 분산 딥러닝 학습과 원격 공유 메모리의 업데이트를 중첩 수행한다(S360).

[0083] 이 후, 원격 공유 메모리에 업데이트되는 전역 학습 카운터가 목표 종료 카운터 이상인지 여부를 판단하고(S365), 전역 학습 카운터가 목표 종료 카운터 이상이면 분산 딥러닝 프로세스를 종료한다(S370).

[0084] 또한, 단계(S365)의 판단결과 전역 학습 카운터가 목표 종료 카운터 미만이면 지속적으로 분산 딥러닝 학습을 수행할 수 있도록 단계(S360)부터 반복 수행할 수 있다.

[0086] 도 6은 본 발명의 일실시예에 따른 분산 딥러닝 장치를 나타낸 블록도이다.

[0087] 도 6을 참조하면, 본 발명의 일실시예에 따른 분산 딥러닝 장치는 프로세서(610) 및 메모리(620)를 포함한다.

[0088] 프로세서(610)는 원격 공유 메모리를 기반으로 전역 파라미터와 전역 학습 카운터를 공유한다.

[0089] 이 때, 원격 공유 메모리에 저장된 전역 파라미터와 전역 학습 카운터는 배타적으로 업데이트가 가능한 데이터

에 상응하는 것으로, 처리 성능이 서로 상이한 복수개의 이기종 분산 딥러닝 장치들이 전체 학습을 효과적으로 나누어 수행할 수 있도록 할 수 있다.

[0090] 이 때, 원격 직접 메모리 접근(REMOTE DIRECT MEMORY ACCESS, RDMA)을 지원하는 고속 네트워크를 기반으로 원격 공유 메모리에 접근할 수 있다. 따라서, 원격 공유 메모리는 프로세서(610)가 전역 파라미터와 전역 학습 카운터에 직접 접근할 수 있도록 지원할 수 있다.

[0091] 또한, 프로세서(610)는 지역 파라미터, 전역-지역 파라미터 차분 및 지역 학습 카운터 영역을 생성한다. 예를 들어, 본 발명의 일실시예에 따른 복수개의 이기종 딥러닝 장치들은 각각 지역 파라미터, 전역-지역 파라미터 차분, 지역 학습 카운터를 포함할 수 있다.

[0092] 이 때, 학습 카운터란, 분산 딥러닝 프로세스들이 분산 딥러닝 학습을 수행할 때 학습한 미니배치(MINI-BATCH)의 전체 횟수를 카운팅하는데 사용될 수 있으며, 각각의 분산 딥러닝 프로세스들이 학습해야 할 미니배치의 순서 번호를 할당 받는데 활용될 수 있다. 이와 같이 각각의 분산 딥러닝 장치로 할당된 학습 카운터는 지역 학습 카운터로써 저장될 수 있다. 이 때, 각각의 분산 딥러닝 장치에 포함된 분산 프로세스들이 수행해야 할 전체 미니배치의 횟수는 분산 딥러닝 프로세스를 이용하는 사용자가 지정할 수 있다.

[0093] 또한, 프로세서(610)는 원격 공유 메모리의 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하는 분산 딥러닝 학습과 원격 공유 메모리 업데이트를 중첩하여 수행한다.

[0094] 이 때, 분산 딥러닝 학습을 통해 지역 파라미터를 자체적으로 학습시킬 수 있고, 원격 공유 메모리에 보관되는 전역 학습 카운터는 메모리(620)에 저장된 지역 학습 카운터와 비교하여 동일하면 변경하는 방식(COMPARE AND SWAP)으로 업데이트될 수 있다.

[0095] 일반적으로 분산 딥러닝 플랫폼에서 분산 딥러닝 학습을 수행하는 프로세스들은 상호간에 대규모 파라미터를 빈번하게 송수신해야 하므로 이 과정에서 발생하는 통신 오버헤드는 전체 분산 딥러닝 학습 성능에서 차지하는 비중이 매우 높은 형편이다. 따라서, 효과적인 분산 딥러닝 학습을 위해서는 통신시간을 감소시키거나 또는 통신시간과 계산시간을 중첩함으로써 통신시간을 숨길 필요가 있다.

[0096] 이와 같은 문제점을 해결하기 위해, 본 발명에서는 학습된 파라미터의 업데이트를 즉각적으로 수행하지 않고 지연하는 방식으로 계산과 통신을 중첩시키는 분산 딥러닝 방법을 제안하고자 한다.

[0097] 이 때, 프로세서(610)는 분산 딥러닝 학습을 위한 딥러닝 학습 스레드(THREAD) 및 원격 공유 메모리 업데이트를 위한 업데이트 스레드(THREAD)를 생성할 수 있다. 일반적으로 메인 스레드가 분산 딥러닝 학습 스레드에 상응할 수 있다.

[0098] 또한, 분산 딥러닝 학습 및 원격 공유 메모리 업데이트는 딥러닝 학습 스레드 및 업데이트 스레드 중 어느 하나로 할당되는 흐름제어락을 기반으로 수행될 수 있다.

[0099] 이하에서는 도 4 내지 도 5를 기반으로 분산 딥러닝 학습과 원격 공유 메모리 업데이트를 중첩 수행하는 두 개의 스레드들의 세부 절차를 설명하도록 한다.

[0100] 먼저, 도 4에 도시된 것처럼, 딥러닝 학습 스레드는 시작되면(S410) 먼저 흐름제어락을 획득할 수 있다(S420).

[0101] 이 때, 흐름제어락은 딥러닝 학습 스레드와 업데이트 스레드 간의 흐름제어를 위해 사용되는 것으로, 흐름제어락은 분산 딥러닝 프로세스가 시작된 이후에 딥러닝 학습 스레드로 먼저 할당될 수 있다.

[0102] 이 후, 전역-지역 파라미터 차분으로부터 지역 파라미터를 업데이트할 수 있다(S430). 이 후, 딥러닝 학습 스레드로 할당되었던 흐름제어락을 해제하고, 업데이트 스레드로 웨이크업 신호를 보내 깨워줄 수 있다(S440). 즉, 분산 딥러닝 학습 스레드는 분산 딥러닝 학습을 시작하기 전에 업데이트 스레드를 깨워준다.

[0103] 이 후, 업데이트된 지역 파라미터를 이용하여 하나의 미니배치 데이터에 대한 분산 딥러닝 학습 수행한 뒤(S450), 학습 결과를 기반으로 지역 파라미터 업데이트할 수 있다(S460).

[0104] 이 후, 원격 공유 메모리에 저장된 전역 학습 카운터 고려하여 추가적인 분산 딥러닝 프로세스가 필요한 경우에는 딥러닝 학습 스레드 반복 수행하되, 전역 학습 카운터가 만료되어 사용자가 지정한 미니배치에 도달하였을 경우에는 딥러닝 분산 프로세스를 종료할 수 있다(S470).

[0105] 또한, 도 5를 참조하면, 업데이트 스레드는 생성된 이후에 딥러닝 학습 스레드 또는 메인 스레드가 깨워줄 때까지 대기할 수 있다(S510).

[0106] 따라서, 웨이크업 신호가 발생하는지 여부를 판단하고(S515), 웨이크업 신호가 발생하여 업데이트 스레드가 깨어나면, 종료변수가 참인지 여부를 확인할 수 있다(S525).

[0107] 단계(S525)의 판단결과 종료변수가 참이면, 업데이트 스레드 종료할 수 있다(S570).

[0108] 또한, 단계(S525)의 판단결과 종료변수가 참이 아니면, 업데이트 스레드로 흐름제어락을 할당할 수 있다(S530).

[0109] 이 후, 원격 공유 메모리에 저장되어 있는 전역 파라미터를 딥러닝 모듈의 지역 버퍼로 읽어와서 전역 파라미터와 지역 파라미터의 차분을 계산할 수 있다(S540).

[0110] 이 후, 단계(S540)을 통해 산출된 전력-지역 파라미터 차분을 이용하여 원격 공유 메모리의 전역 파라미터가 증가하도록 업데이트한 뒤(S550) 업데이트 스레드로 할당된 흐름제어락을 해제할 수 있다(S560).

[0111] 이 때, 단계(S530) 내지 단계(S560)의 절차는 분산 딥러닝 학습이 완료될 때까지 반복적으로 수행될 수 있으며, 대체로 분산 딥러닝 학습시간이 원격 공유 메모리 업데이트 시간보다 길기 때문에 업데이트 스레드는 전역 파라미터 업데이트 완료 후에 대기상태로 회귀할 수 있다.

[0112] 이와 같이, 본 발명에서는 분산 딥러닝 장치가 N번째로 학습한 파라미터가 N+1번째 학습 도중에 전역 파라미터로 업데이트될 수 있고, N번째 학습 도중에 읽어온 전역 파라미터를 N+1번째 학습 전에 지역 파라미터로써 업데이트하여 학습을 수행할 수 있다. 따라서, 종래에 파라미터 서버를 이용하는 비동기 방식처럼 새로운 전역 파라미터가 업데이트될 때까지 대기할 필요가 없으므로 통신에 의해 지체되는 시간을 절약할 수 있다. 즉, 본 발명에서는 딥러닝 학습 스레드와 업데이트 스레드가 흐름제어락과 대기/웨이크업 방식을 이용하여 학습과 통신(전역 파라미터 업데이트)을 중첩 수행할 수 있다.

[0113] 또한, 프로세서(610)는 원격 공유 메모리 업데이트를 기반으로 업데이트된 전역 학습 카운터를 고려하여 분산 딥러닝 프로세스를 종료한다.

[0114] 예를 들어, 프로세서(610)는 원격 공유 메모리에 저장된 전역 학습 카운터를 배타적으로 증가시키면서 분산 딥러닝 학습을 수행하므로, 전역 학습 카운터가 사용자가 지정한 값에 도달하였을 때에 분산 딥러닝 프로세스를 종료할 수 있다.

[0115] 이와 같이 함으로써 저속 GPU는 전체 미니배치 횟수 중에서 더 적은 횟수를 학습하고, 고속 GPU는 더 많은 미니배치를 학습할 수 있으므로 분산 딥러닝 프로세스의 사용자가 지정한 미니배치에 도달하였을 때에 복수개의 분산 딥러닝 장치들은 거의 동시에 분산 딥러닝 학습을 종료할 수 있다.

[0116] 또한, 프로세서(610)는 원격 공유 메모리에 전역 파라미터 영역 및 전역 학습 카운터 영역을 생성한다.

[0117] 예를 들어, 프로세서(610)는 RDMA 기반의 고속 네트워크를 기반으로 원격 공유 메모리로 접근하여 전역 파라미터 영역과 전역 학습 카운터 영역을 생성할 수 있다.

[0118] 또한, 프로세서(610)는 전역 파라미터 및 전역 학습 카운터를 초기화한다.

[0119] 이 때, 전역 파라미터는 분산 딥러닝 장치 별로 다양한 방식으로 초기화될 수 있으며, 전역 학습 카운터는 0으로 초기화될 수 있다.

[0120] 또한, 이와 같은 초기화 과정은 업데이트 스레드가 최초로 웨이크업되는 시점을 기준으로 수행될 수도 있다.

[0121] 메모리(620)는 지역 파라미터, 전역-지역 파라미터 차분 및 지역 학습 카운터를 저장한다.

[0122] 또한, 메모리(620)는 상술한 바와 같이 본 발명의 실시예에 따른 이종 클러스터 기반의 분산 딥러닝 과정에서 발생되는 다양한 정보를 저장한다.

[0123] 실시예에 따라, 메모리(620)는 분산 딥러닝 장치와 독립적으로 구성되어 분산 딥러닝 수행을 위한 기능을 지원할 수 있다. 이 때, 메모리(620)는 별도의 대용량 스토리지로 동작할 수 있고, 동작 수행을 위한 제어 지능을 포함할 수 있다.

[0124] 한편, 분산 딥러닝 장치는 메모리가 탑재되어 그 장치 내에서 정보를 저장할 수 있다. 일 구현예의 경우, 메모리는 컴퓨터로 판독 가능한 매체이다. 일 구현 예에서, 메모리는 휘발성 메모리 유닛일 수 있으며, 다른 구현예의 경우, 메모리는 비휘발성 메모리 유닛일 수도 있다. 일 구현예의 경우, 저장장치는 컴퓨터로 판독 가능한 매체이다. 다양한 서로 다른 구현 예에서, 저장장치는 예컨대 하드디스크 장치, 광학디스크 장치, 혹은 어떤 다른 대용량 저장장치를 포함할 수도 있다.

[0125] 이와 같은 분산 딥러닝 장치를 이용함으로써 이기종의 GPU를 이용한 분산 딥러닝 수행 시 통신 오버헤드를 감소시킬 수 있다.

[0126] 또한, 동시에 성능이 다른 GPU들을 효과적으로 사용할 수 있는 분산 딥러닝 방법을 제공할 수 있다.

[0127] 또한, 학습 속도가 다른 분산 프로세스들이 전체 학습을 효과적으로 나누어 수행할 수 있도록 할 수 있다.

[0128] 또한, 학습한 파라미터의 업데이트를 지연하는 방식으로 계산과 통신을 중첩하여 각각의 GPU 활용률을 극대화함으로써 우수한 분산 처리 확장성을 제공할 수 있다.

[0130] 이상에서와 같이 본 발명에 따른 이종 클러스터 기반의 분산 딥러닝 방법 및 이를 위한 장치는 상기한 바와 같이 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상기 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

부호의 설명

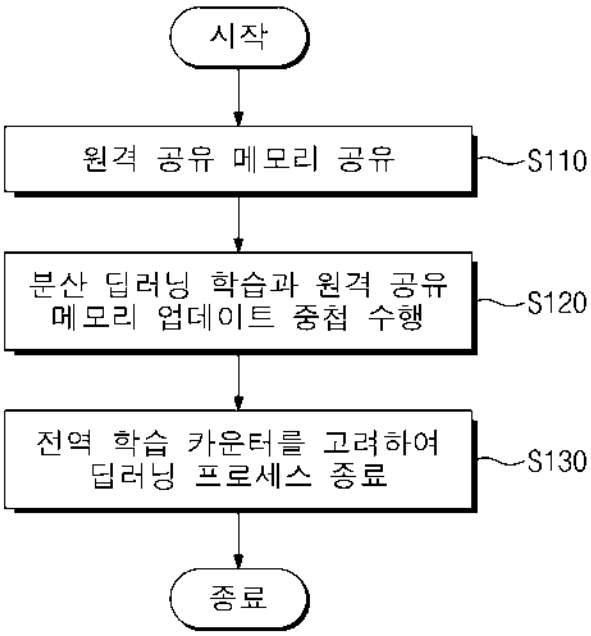
[0131] 210-1~210-N: 딥러닝 모듈 220: 원격 공유 메모리

230: RDMA 고속 네트워크 610: 프로세서

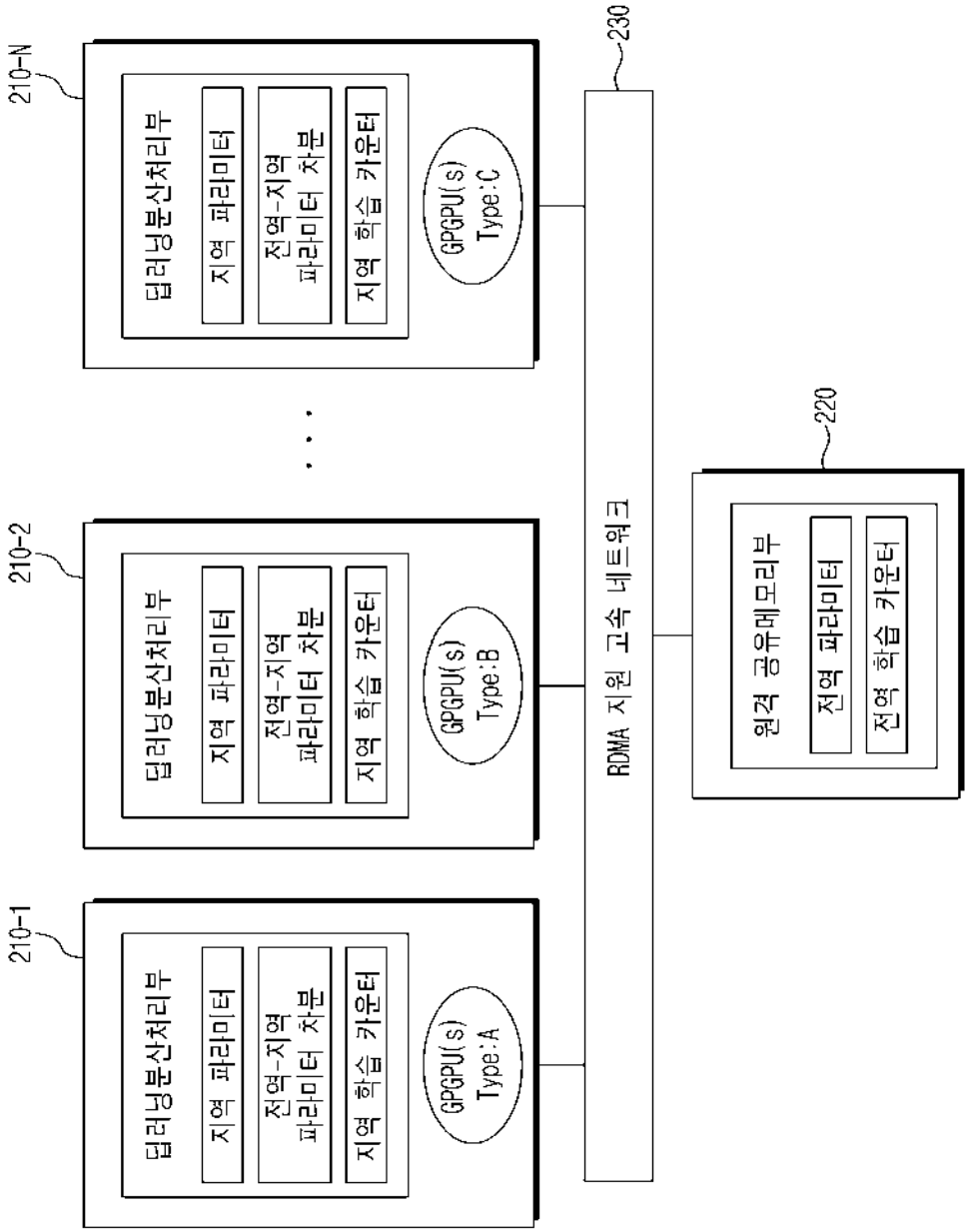
620: 메모리

도면

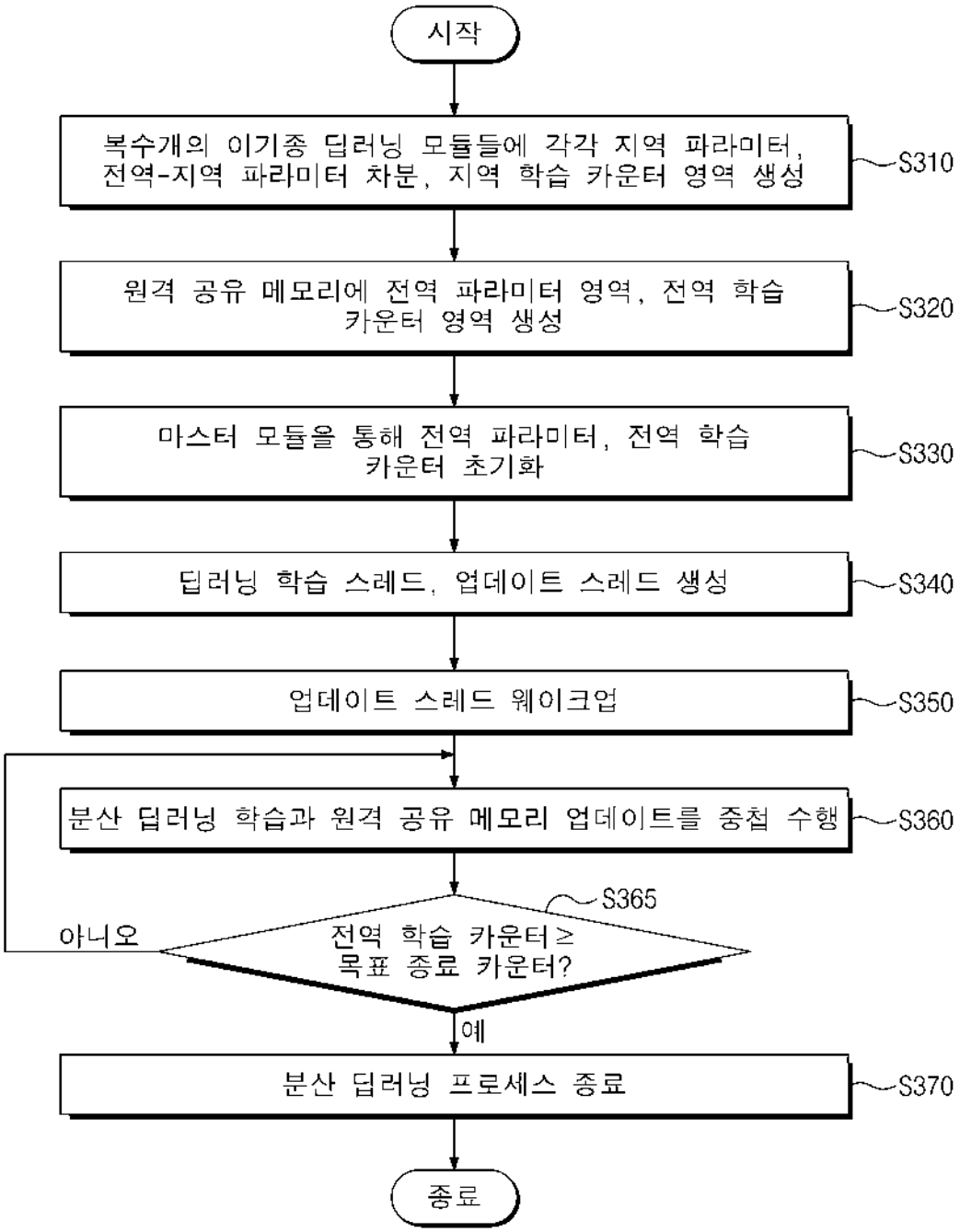
도면1



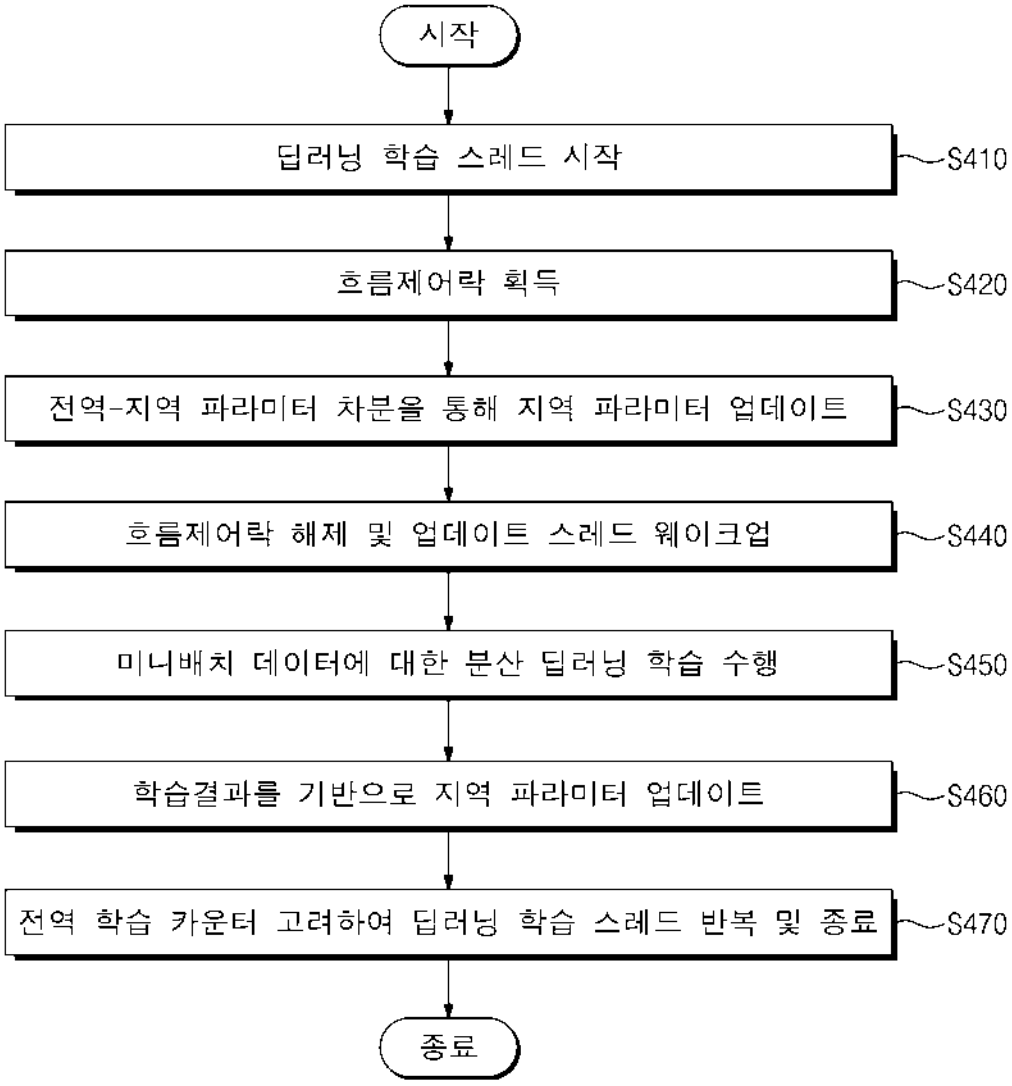
도면2


210-N
딥러닝분산처리부
지역 파라미터
전역-지역 파라미터 차분
지역 학습 카운터
GPGPU(s) Type:C
210-2
딥러닝분산처리부
지역 파라미터
전역-지역 파라미터 차분
지역 학습 카운터
GPGPU(s) Type:B
210-1
딥러닝분산처리부
지역 파라미터
전역-지역 파라미터 차분
지역 학습 카운터
GPGPU(s) Type:A
230
RDMA 지원 고속 네트워크
220
원격 공유메모리부
전역 파라미터
전역 학습 카운터

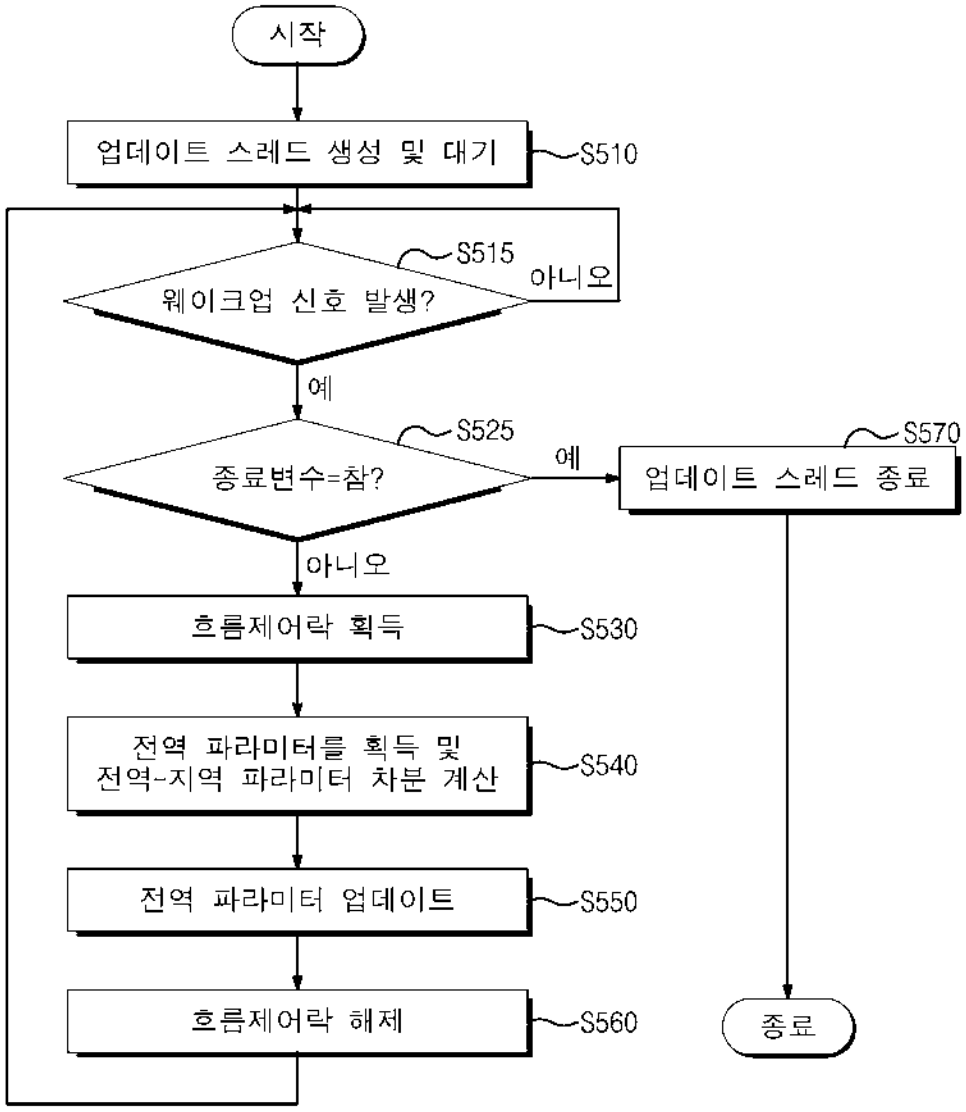
도면3


시작
복수개의 이기종 딥러닝 모듈들에 각각 지역 파라미터, 전역-지역 파라미터 차분, 지역 학습 카운터 영역 생성
S310
원격 공유 메모리에 전역 파라미터 영역, 전역 학습 카운터 영역 생성
S320
마스터 모듈을 통해 전역 파라미터, 전역 학습 카운터 초기화
S330
딥러닝 학습 스레드, 업데이트 스레드 생성
S340
업데이트 스레드 웨이크업
S350
분산 딥러닝 학습과 원격 공유 메모리 업데이트를 중첩 수행
S360
S365
전역 학습 카운터≥ 목표 종료 카운터?
아니오
예
분산 딥러닝 프로세스 종료
S370
종료

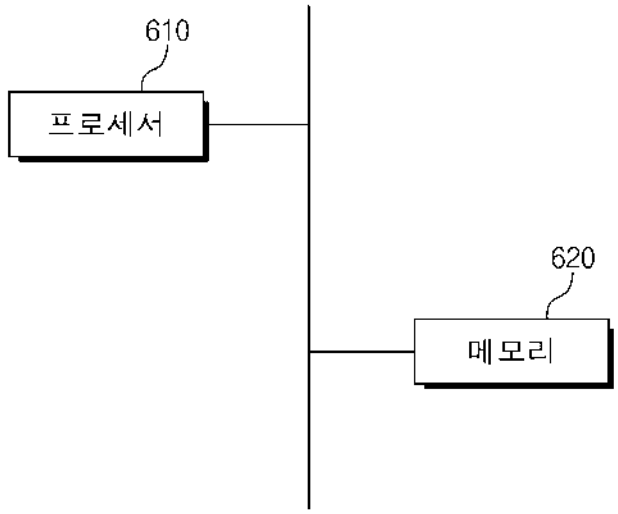
도면4


시작
딥러닝 학습 스레드 시작
S410
흐름제어락 획득
S420
전역-지역 파라미터 차분을 통해 지역 파라미터 업데이트
S430
흐름제어락 해제 및 업데이트 스레드 웨이크업
S440
미니배치 데이터에 대한 분산 딥러닝 학습 수행
S450
학습결과를 기반으로 지역 파라미터 업데이트
S460
전역 학습 카운터 고려하여 딥러닝 학습 스레드 반복 및 종료
S470
종료

도면5


시작
업데이트 스레드 생성 및 대기
S510
S515
웨이크업 신호 발생?
아니오
예
S525
종료변수=참?
예
S570
업데이트 스레드 종료
아니오
흐름제어락 획득
S530
전역 파라미터를 획득 및
전역-지역 파라미터 차분 계산
S540
전역 파라미터 업데이트
S550
흐름제어락 해제
S560
종료

도면6



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 9

【변경전】

원격 공유 메모리의 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하도록 기저장된 지역 파라미터와 상기 전역 파라미터의 차분 연산 결과를 이용하여 분산 딥러닝 학습을 수행하고,

상기 분산 딥러닝 학습이 수행되는 동안 상기 차분 연산 결과를 이용하여 상기 전역 파라미터를 업데이트하는 분산 딥러닝 프로세스를 수행하고,

상기 분산 딥러닝 학습의 수행 횟수에 상응하는 상기 지역 학습 카운터를 기반으로 상기 전역 학습 카운터를 업데이트하고, 상기 업데이트된 전역 학습 카운터가 기설정된 종료 카운터 이상인지 여부를 판단하여 상기 분산 딥러닝 프로세스를 종료하는 프로세서; 및

상기 지역 파라미터, 상기 전역 파라미터와 상기 지역 파라미터의 차분 연산 결과 및 상기 지역 학습 카운터를 저장하는 메모리

를 포함하는 것을 특징으로 하는 분산 딥러닝 장치.

【변경후】

원격 공유 메모리의 전역 학습 카운터를 기반으로 할당된 지역 학습 카운터에 상응하도록 기저장된 지역 파라미터와 전역 파라미터의 차분 연산 결과를 이용하여 분산 딥러닝 학습을 수행하고,

상기 분산 딥러닝 학습이 수행되는 동안 상기 차분 연산 결과를 이용하여 상기 전역 파라미터를 업데이트하는 분산 딥러닝 프로세스를 수행하고,

상기 분산 딥러닝 학습의 수행 횟수에 상응하는 상기 지역 학습 카운터를 기반으로 상기 전역 학습 카운터를 업데이트하고, 상기 업데이트된 전역 학습 카운터가 기설정된 종료 카운터 이상인지 여부를 판단하여 상기 분산 딥러닝 프로세스를 종료하는 프로세서; 및

상기 지역 파라미터, 상기 전역 파라미터와 상기 지역 파라미터의 차분 연산 결과 및 상기 지역 학습 카운터를 저장하는 메모리

를 포함하는 것을 특징으로 하는 분산 딥러닝 장치.

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0125734
(43) 공개일자 2018년11월26일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2006.01) *G06F 12/02* (2018.01)
G06N 3/08 (2006.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06F 12/0292 (2013.01)
(21) 출원번호 10-2017-0060400
(22) 출원일자 2017년05월16일
심사청구일자 없음

(71) 출원인
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
(72) 발명자
임은지
대전광역시 유성구 노은동로 187, 602동 1801호
안신영
대전광역시 서구 둔산북로 160, 5동 701호
(뒷면에 계속)
(74) 대리인
한양특허법인

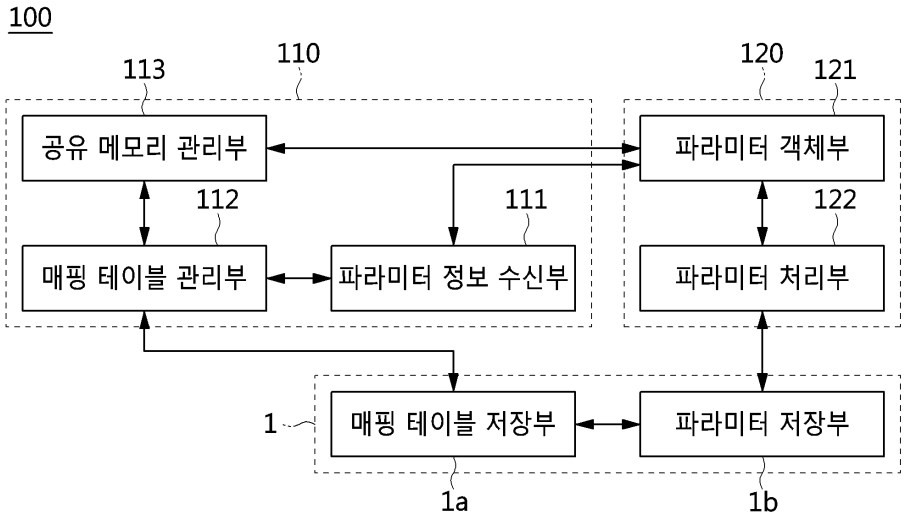
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **파라미터 공유 장치 및 방법**

(57) 요 약

파라미터 공유 장치 및 방법이 개시된다. 본 발명의 일실시예에 따른 파라미터 공유 장치는 메모리 박스에 파라미터가 저장될 메모리 영역의 할당 관리를 수행하고, 상기 메모리 영역의 할당 관리에 따라 상기 메모리 박스에 저장된 매핑 테이블을 업데이트 하는 메모리 할당부 및 상기 파라미터가 저장될 메모리 영역의 할당 관리를 위한 파라미터 정보를 상기 메모리 할당부에 제공하고, 상기 메모리 박스에 저장된 파라미터를 공유하는 연산 처리부를 포함한다.

대 표 도 - 도5



(52) CPC특허분류
G06N 3/08 (2013.01)
(72) 발명자
최용석
전광역시 유성구 지족북로 60, 207동 303호
우영춘
대전광역시 유성구 어은로 57, 113동 404호
최완
대전광역시 서구 관저북로 52, 108동 306호

이 발명을 지원한 국가연구개발사업
과제고유번호 R7117-16-0235
부처명 미래창조과학부
연구관리전문기관 정보통신기술진흥센터(IITP)
연구사업명 정보통신방송기술개발사업(SW컴퓨팅 산업원천기술개발사업)
연구과제명 대규모 딥러닝 고속 처리를 위한 HPC 시스템 개발
기 여 율 1/1
주관기관 한국전자통신연구원
연구기간 2016.04.01 ~ 2016.12.31

명 세 서

청구범위

청구항 1

파라미터 공유 장치를 이용하는 방법에 있어서,

메모리 박스에 저장될 파라미터의 메모리 영역을 할당하기 위한 파라미터 정보를 수신하는 단계;

상기 메모리 박스의 매핑 테이블에 잠금을 걸고, 매핑 테이블을 읽어오는 단계;

상기 파라미터 정보에 기반하여 상기 매핑 테이블에서 상기 메모리 박스에 파라미터를 저장할 메모리 영역의 할당 여부를 확인하는 단계;

상기 메모리 영역의 할당 여부에 따라 매핑 정보를 수정한 매핑 테이블을 상기 메모리 박스에 쓰고, 상기 매핑 테이블의 잠금을 해제하는 단계; 및

상기 메모리 영역을 할당한 메모리 주소를 고려하여 상기 파라미터를 공유하는 단계;

를 포함하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 2

청구항 1에 있어서,

상기 수신하는 단계는

상기 파라미터의 파라미터 식별자 및 상기 파라미터 크기 중 적어도 하나를 포함하는 파라미터 정보를 수신하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 3

청구항 2에 있어서,

상기 매핑 테이블은

각각의 엔트리에 파라미터의 식별자, 메모리 영역에 대한 메모리 주소 및 참조 카운트를 포함하는 매핑 정보가 기록된 것을 특징으로 하는 파라미터 공유 방법.

청구항 4

청구항 3에 있어서,

상기 할당 여부를 확인하는 단계는

상기 매핑 테이블의 엔트리를 확인하여 상기 메모리 박스에 상기 파라미터의 메모리 영역이 기 할당되어 있는 경우,

상기 매핑 테이블에 상기 파라미터에 상응하는 엔트리에 참조 카운트를 증가시켜 상기 매핑 테이블을 업데이트하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 5

청구항 4에 있어서,

상기 할당 여부를 확인하는 단계는

상기 매핑 테이블의 엔트리를 확인하여 상기 메모리 박스에 상기 파라미터의 메모리 영역이 할당되지 않은 경우,

상기 파라미터 크기만큼 상기 메모리 박스에 메모리 영역을 할당하고, 상기 메모리 영역이 할당된 파라미터에 대한 매핑 정보를 상기 매핑 테이블의 새로운 엔트리에 추가하여 상기 매핑 테이블을 업데이트하는 것을 특징으

로 하는 파라미터 공유 방법.

청구항 6

청구항 5에 있어서,

상기 매핑 테이블의 잠금을 해제하는 단계는

상기 메모리 영역이 할당된 파라미터에 대한 상기 메모리 박스의 메모리 주소를 상기 파라미터 공유 장치에 기록하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 7

청구항 6에 있어서,

상기 공유하는 단계는

상기 파라미터 공유 장치에 기록된 메모리 주소를 참조하여 상기 메모리 박스에 저장된 파라미터 값을 읽어오는(read) 단계;

모델 알고리즘을 이용하여 상기 메모리 박스의 파라미터 값에 상응하는 파라미터 차분 값을 계산하는 단계; 및

상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 파라미터 값을 수정하는 단계;

를 포함하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 8

청구항 7에 있어서,

상기 파라미터 값을 수정하는 단계는

상기 메모리 박스가 합저장 기능 수행이 가능한 경우,

상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 합저장 기능을 통해 상기 메모리 박스의 파라미터 값을 수정하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 9

청구항 7에 있어서,

상기 파라미터 값을 수정하는 단계는

상기 메모리 박스가 합저장 기능 수행이 불가능한 경우,

상기 메모리 박스로부터 파라미터 값을 다시 읽어오고(read), 상기 파라미터 차분 값과 다시 읽어온 파라미터 값을 이용하여 산출한 파라미터 수정 값을 상기 메모리 박스에 쓰는(write) 것을 특징으로 하는 파라미터 공유 방법.

청구항 10

파라미터 공유 장치를 이용하는 방법에 있어서,

메모리 박스에 파라미터가 저장된 메모리 영역을 해제하기 위한 파라미터 정보를 수신하는 단계;

상기 메모리 박스의 매핑 테이블에 잠금을 걸고, 매핑 테이블을 읽어오는 단계;

상기 매핑 테이블에 기반하여 상기 메모리 박스에 상기 파라미터가 할당된 메모리 영역의 해제 여부를 확인하는 단계;

상기 메모리 영역의 해제 여부에 따라 매핑 정보를 수정한 매핑 테이블을 상기 메모리 박스에 쓰고, 상기 매핑 테이블의 잠금을 해제하는 단계; 및

상기 메모리 영역을 해제한 메모리 주소를 고려하여 상기 파라미터를 공유하는 단계;

를 포함하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 11

청구항 10에 있어서,

상기 수신하는 단계는

상기 파라미터의 파라미터 식별자 및 상기 파라미터가 저장된 메모리 영역에 대한 메모리 주소 중 적어도 하나를 포함하는 파라미터 정보를 수신하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 12

청구항 11에 있어서,

상기 읽어오는 단계는

상기 매핑 테이블에 상기 파라미터에 상응하는 엔트리에 참조 카운트를 감소시켜 상기 매핑 테이블을 업데이트하는 단계; 및

상기 파라미터 공유 장치에 기록된 상기 파라미터에 상응하는 메모리 주소를 삭제하는 단계를 포함하는 파라미터 공유 방법.

청구항 13

청구항 12에 있어서,

상기 해제 여부를 확인하는 단계는

상기 매핑 테이블의 참조 카운트가 최소값인 경우, 상기 메모리 박스에 할당된 메모리 영역을 해제하고, 상기 메모리 영역에 상응하는 매핑 테이블의 엔트리를 삭제하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 14

청구항 13에 있어서,

상기 공유하는 단계는

상기 파라미터 공유 장치에 기록된 메모리 주소를 참조하여 상기 메모리 박스에 저장된 파라미터 값을 읽어오는(read) 단계;

모델 알고리즘을 이용하여 상기 메모리 박스의 파라미터 값에 상응하는 파라미터 차분 값을 계산하는 단계; 및

상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 파라미터 값을 수정하는 단계;

를 포함하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 15

청구항 14에 있어서,

상기 파라미터 값을 수정하는 단계는

상기 메모리 박스가 합저장 기능 수행이 가능한 경우,

상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 합저장 기능을 통해 상기 메모리 박스의 파라미터 값을 수정하는 것을 특징으로 하는 파라미터 공유 방법.

청구항 16

청구항 14에 있어서,

상기 파라미터 값을 수정하는 단계는

상기 메모리 박스가 합저장 기능 수행이 불가능한 경우,

상기 메모리 박스로부터 파라미터 값을 다시 읽어오고(read), 상기 파라미터 차분 값과 다시 읽어온 파라미터 값을 이용하여 산출한 파라미터 수정 값을 상기 메모리 박스에 쓰는(write) 것을 특징으로 하는 파라미터 공유

방법.

청구항 17

메모리 박스에 파라미터가 저장될 메모리 영역의 할당 관리를 수행하고, 상기 메모리 영역의 할당 관리에 따라 상기 메모리 박스에 저장된 매핑 테이블을 업데이트 하는 메모리 할당부; 및

상기 파라미터가 저장될 메모리 영역의 할당 관리를 위한 파라미터 정보를 상기 메모리 할당부에 제공하고, 상기 메모리 박스에 저장된 파라미터를 공유하는 연산 처리부;

를 포함하는 것을 특징으로 하는 파라미터 공유 장치.

청구항 18

청구항 17에 있어서,

상기 메모리 할당부는

상기 연산 처리부로부터 상기 메모리 영역의 할당 관리를 위한 상기 파라미터 정보를 수신하는 파라미터 정보 수신부;

상기 메모리 박스에 저장된 매핑 테이블의 잠금을 관리하고, 상기 매핑 테이블을 업데이트하는 매핑 테이블 관리부; 및

상기 매핑 테이블의 참조 카운트를 수정하여, 상기 메모리 영역의 할당 관리를 수행하는 공유 메모리 관리부;

를 포함하는 것을 특징으로 하는 파라미터 공유 장치.

청구항 19

청구항 18에 있어서,

상기 메모리 할당부는

상기 참조 카운트에 따라 상기 메모리 영역에서 파라미터를 공유 중인 다른 파라미터 공유 장치의 개수를 확인하는 것을 특징으로 하는 파라미터 공유 장치.

청구항 20

청구항 19에 있어서,

상기 메모리 박스는

상기 연산 처리부의 파라미터 값 수정 요청에 따라 합저장 기능을 이용하여 상기 메모리 박스에 저장된 파라미터 값을 업데이트하는 것을 특징으로 하는 파라미터 공유 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 파라미터 기술에 관한 것으로, 보다 상세하게는 분산 딥러닝을 위한 워커 장치들이 딥러닝 모델 파라미터를 공유하기 위한 기술에 관한 것이다.

배 경 기 술

[0002] 최근 이미지 인식, 음성 인식, 자연어 처리의 발전에 기여하며 주목 받고 있는 딥러닝 모델은 사람의 신경세포(Biological Neuron)를 모사하여 기계가 학습하도록 하는 인공신경망 (Artificial Neural Network) 기반의 기계 학습법이다.

[0003] 최근 딥러닝 모델들은 응용의 인식 성능을 높이기 위해 모델의 계층이 깊어지고(Deep), 특징(피쳐)이 많아지는 대규모 모델로 진화하고 있다. 딥러닝 모델의 규모가 커지고 입력 데이터의 양이 많아질수록 학습할 파라미터가 많아지고 계산량도 늘어난다. 이에 따라 많은 컴퓨터가 필요하며 분산 시스템에서 병렬적으로 연산하면 학습을 가속화 할 수 있다.

[0004] 딥러닝 학습의 분산 병렬 처리 시에 매 학습과정이 반복될 때마다 각 분산 컴퓨터 장치(워커 장치)들이 계산한 파라미터를 서로 공유할 수 있다. 많은 분산 딥러닝 시스템에서는 중앙 집중형 파라미터 서버를 이용하여 파라미터를 공유한다. 파라미터 서버는 매 학습과정 마다 각 워커 장치에서 학습된 파라미터를 수집하고 종합하여 다시 워커 장치들에게 나누어주는 역할을 한다.

[0005] 분산 학습된 파라미터를 업데이트하는 시점에 따라서 동기식 방식과 비동기식 방식으로 나눌 수 있다. 동기식 업데이트의 경우는 모든 워커 장치가 한 학습과정을 마친 시점에 파라미터 서버와 통신하여 파라미터를 업데이트 하는 방식으로써, 여러 워커 장치들 중에서 연산 속도가 가장 느린 워커 장치에 의해서 전체 학습 성능이 결정되게 된다. 비동기식 업데이트 방식은 파라미터 서버가 컴퓨터들로부터 늦거나 빨리 도착하는 파라미터들의 동기를 맞추지 않고 학습을 진행하는 방식이다. 비동기식 방식은 동기식 방식에 비해 정확성을 크게 희생시키지 않으면서 빠르게 트레이닝 할 수 있는 장점이 있어서, 최근 분산 딥러닝 학습에서는 비동기식 방식을 많이 채택하고 있다.

[0006] 분산 딥러닝 학습에서 워커 장치가 많아질수록 병렬도가 높아지므로 연산 속도는 빨라지지만 연산된 결과를 파라미터 서버와 통신하는데 걸리는 시간이 상대적으로 늘어나게 된다. 파라미터 서버와의 통신 속도가 느릴 경우에 전체 학습 성능이 저하될 수 있다. 따라서 분산 병렬 환경에서 딥러닝 모델을 학습할 때 파라미터 교환 시간이 중요한 요소라고 볼 수 있다.

[0007] 한편, 한국공개특허 제 10-2012-0140104 호"메모리의 데이터 저장 방법"는 메모리의 데이터 저장 방법에 관한 것으로서, 더욱 상세하게는 차량의 제어기 등에서 각 변수 조건에 따라 연산된 학습치를 메모리 영역에 효율적으로 저장할 수 있는 메모리의 데이터 저장 방법에 관하여 개시하고 있다.

[0008] 그러나, 한국공개특허 제 10-2012-0140104 호는 학습치(파라미터)를 메모리에 효율적으로 저장하기 위한 것으로, 메모리에 저장된 파라미터를 다수의 워커 장치들이 효과적으로 공유하는 측면에 대해서는 침묵하고 있다.

발명의 내용

해결하려는 과제

[0009] 본 발명은 분산 딥러닝 학습에서 다수의 워커 장치들 간의 파라미터 공유를 위해서 파라미터 서버를 사용하는 대신에, 공유 메모리 장치인 메모리 박스의 공유 메모리를 통해서 파라미터를 공유하도록 하여 딥러닝 학습을 가속화하는 것을 목적으로 한다.

[0010] 파라미터 서버와 워커들은 컴퓨터 간의 통신 네트워크(예를 들어, 이더넷(Ethernet))를 통해서 요청-응답(request-response) 방식으로 파라미터를 송수신한다. 다시 말해서, 워커가 파라미터 값을 필요로 할 때는, 파라미터 서버에게 파라미터 값에 대한 요청 메시지를 전송하고, 파라미터 서버는 자신의 메인 메모리로부터 파라미터 값을 읽어와서 요청에 대한 응답으로 워커에게 전송한다. 반대로, 워커가 파라미터 값을 업데이트하고자 할 때는, 파라미터 차분값 또는 파라미터 수정값을 포함한 파라미터 업데이트 요청 메시지를 파라미터 서버에게 전송하고, 파라미터 서버는 받은 값을 이용하여 메인 메모리에 저장된 파라미터의 값을 업데이트하고 워커에게 응답 메시지를 전송한다.

[0011] 분산 딥러닝 학습을 수행할 때 다수의 분산 워커 간에 대규모 파라미터 송수신이 빈번하게 발생하는데, 상기에서 기술한 방식대로 파라미터 서버를 이용하면 네트워크를 통한 통신 오버헤드가 크게 발생하고, 워커와 파라미터 서버에서 메시지 처리 시간도 크게 나타날 수 있다. 따라서, 이보다 개선된 방식이 필요하다.

[0012] 그에 반해서 메모리 박스는 독립적인(stand-alone) 컴퓨터가 아니고, 컴퓨터에 장착하여 사용 할 수 있는 하나의 장치(device)이다. 메모리 박스는 대용량의 메모리를 보유하고 PCIe와 같은 시스템 버스를 통해서 컴퓨터에 연결된다. 따라서, 파라미터 서버에 비해서 매우 빠른 속도로 데이터를 제공할 수 있다. 또한, 메모리 박스는 다수의 연결 커넥터를 보유하고 있어서, 동시에 다수의 워커와 연결되어 그들로부터 공유될 수 있다. 메모리 박스가 보유한 대규모 메모리는 다수의 워커들이 공유 메모리로 사용할 수 있다.

[0013] 파라미터 서버와 메모리 박스는 이러한 차이점으로 인해서 사용 방법이 크게 다르다. 메모리 박스는 컴퓨터 장치이므로 이를 사용할 때 워커가 주도적(active)으로 동작한다. 다시 말해서, 워커가 메모리 박스로부터 데이터를 리드(read)하여 파라미터 값을 가져 가고, 반대로 메모리 박스에 데이터를 라이트(write)하여 파라미터 값을 저장할 수 있다. 또한, 분산 워커들이 공유 메모리를 활용하여 딥러닝 파라미터를 공유하기 위해서는 새로운 파

라미터 공유 방법이 필요하다.

[0014] 이러한 특징들로 인하여 기존의 파라미터 서버를 사용하던 분산 딥러닝 프레임워크로는 메모리 박스를 이용할 수 없다. 메모리 박스를 이용하여 파라미터를 공유하면 메모리 박스의 빠른 접근 속도로 인해서 딥러닝 학습을 가속화 할 수 있다. 그러나, 메모리 박스를 이용하기 위해서는 분산 딥러닝 프레임워크가 메모리 박스를 통하여 파라미터를 공유하도록 수정되어야 한다.

[0015] 따라서, 상기와 같은 이유로 인해서 본 발명의 목적은, 분산 딥러닝 학습에서 다수의 워커 장치들이 메모리 박스의 공유 메모리를 통해서 파라미터를 공유할 수 있는, 파라미터 공유 장치 및 방법을 제공하는데 있다.

[0016] 또한, 본 발명은 분산 딥러닝 학습에서 파라미터 서버를 메모리 박스로 대체 지원함에 있어서 딥러닝 프레임워크가 가진 원래의 기능과 사용자가 사용하는 딥러닝 모델 개발 및 학습 인터페이스에 수정을 가하지 않고, 다수의 워커 장치가 투명하게 메모리 박스를 통해 파라미터를 공유하는 것을 목적으로 한다.

과제의 해결 수단

[0017] 상기한 목적을 달성하기 위한 본 발명의 일실시예에 따른 파라미터 공유 방법은 파라미터 공유 장치를 이용하는 방법에 있어서, 메모리 박스에 저장될 파라미터의 메모리 영역을 할당하기 위한 파라미터 정보를 수신하는 단계; 상기 메모리 박스의 매핑 테이블에 잠금을 걸고, 매핑 테이블을 읽어오는 단계; 상기 파라미터 정보에 기반하여 상기 매핑 테이블에서 상기 메모리 박스에 파라미터를 저장할 메모리 영역의 할당 여부를 확인하는 단계; 상기 메모리 영역의 할당 여부에 따라 매핑 정보를 수정한 매핑 테이블을 상기 메모리 박스에 쓰고, 상기 매핑 테이블의 잠금을 해제하는 단계 및 상기 메모리 영역을 할당한 메모리 주소를 고려하여 상기 파라미터를 공유하는 단계를 포함한다.

[0018] 이 때, 상기 수신하는 단계는 상기 파라미터의 파라미터 식별자 및 파라미터 크기 중 적어도 하나를 포함하는 파라미터 정보를 수신할 수 있다.

[0019] 이 때, 파라미터 크기는 파라미터를 저장하기 위해 필요한 메모리 크기일 수 있다.

[0020] 이 때, 상기 매핑 테이블은 각각의 엔트리에 파라미터의 식별자, 메모리 영역에 대한 메모리 주소 및 참조 카운트를 포함하는 매핑 정보가 기록될 수 있다.

[0021] 이 때, 상기 할당 여부를 확인하는 단계는 상기 매핑 테이블의 엔트리를 확인하여 상기 메모리 박스에 상기 파라미터의 메모리 영역이 기 할당되어 있는 경우, 상기 매핑 테이블에 상기 파라미터에 상응하는 엔트리에 참조 카운트를 증가시켜 상기 매핑 테이블을 업데이트 할 수 있다.

[0022] 이 때, 상기 할당 여부를 확인하는 단계는 상기 매핑 테이블의 엔트리를 확인하여 상기 메모리 박스에 상기 파라미터의 메모리 영역이 할당되지 않은 경우, 상기 파라미터 크기만큼 상기 메모리 박스에 메모리 영역을 할당하고, 상기 메모리 영역이 할당된 파라미터에 대한 매핑 정보를 상기 매핑 테이블의 새로운 엔트리에 추가하여 상기 매핑 테이블을 업데이트 할 수 있다.

[0023] 이 때, 상기 매핑 테이블의 잠금을 해제하는 단계는 상기 메모리 영역이 할당된 파라미터에 대한 상기 메모리 박스의 메모리 주소를 상기 파라미터 공유 장치에 기록할 수 있다.

[0024] 이 때, 상기 공유하는 단계는 상기 파라미터 공유 장치에 기록된 메모리 주소를 참조하여 상기 메모리 박스에 저장된 파라미터 값을 읽어오는(read) 단계; 모델 알고리즘을 이용하여 상기 메모리 박스의 파라미터 값에 상응하는 파라미터 차분 값을 계산하는 단계 및 상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 파라미터 값을 수정하는 단계를 포함할 수 있다.

[0025] 이 때, 상기 파라미터 값을 수정하는 단계는 상기 메모리 박스가 합저장 기능 수행이 가능한 경우, 상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 합저장 기능을 통해 상기 메모리 박스의 파라미터 값을 수정할 수 있다.

[0026] 이 때, 상기 파라미터 값을 수정하는 단계는 상기 메모리 박스가 합저장 기능 수행이 불가능한 경우, 상기 메모리 박스로부터 파라미터 값을 다시 읽어오고(read), 상기 파라미터 차분 값과 다시 읽어온 파라미터 값을 이용하여 산출한 파라미터 수정 값을 상기 메모리 박스에 쓸 수 있다(write).

[0027] 또한, 상기한 목적을 달성하기 위한 본 발명의 일실시예에 따른 파라미터 공유 방법은 파라미터 공유 장치를 이용하는 방법에 있어서, 메모리 박스에 파라미터가 저장된 메모리 영역을 해제하기 위한 파라미터 정보를 수신하

는 단계; 상기 메모리 박스의 매핑 테이블에 잠금을 걸고, 매핑 테이블을 읽어오는 단계; 상기 매핑 테이블에 기반하여 상기 메모리 박스에 상기 파라미터가 할당된 메모리 영역의 해제 여부를 확인하는 단계; 상기 메모리 영역의 해제 여부에 따라 매핑 정보를 수정한 매핑 테이블을 상기 메모리 박스에 쓰고, 상기 매핑 테이블의 잠금을 해제하는 단계 및 상기 메모리 영역을 해제한 메모리 주소를 고려하여 상기 파라미터를 공유하는 단계를 포함한다.

[0028] 이 때, 상기 수신하는 단계는 상기 파라미터의 파라미터 식별자 및 상기 파라미터가 저장된 메모리 영역에 대한 메모리 주소 중 적어도 하나를 포함하는 파라미터 정보를 수신할 수 있다.

[0029] 이 때, 상기 매핑 테이블은 각각의 엔트리에 파라미터의 식별자, 메모리 영역에 대한 메모리 주소 및 참조 카운트를 포함하는 매핑 정보가 기록될 수 있다.

[0030] 이 때, 상기 읽어오는 단계는 상기 매핑 테이블에 상기 파라미터에 상응하는 엔트리에 참조 카운트를 감소시켜 상기 매핑 테이블을 업데이트 하는 단계 및 상기 파라미터 공유 장치에 기록된 상기 파라미터에 상응하는 메모리 주소를 삭제하는 단계를 포함할 수 있다.

[0031] 이 때, 상기 해제 여부를 확인하는 단계는 상기 매핑 테이블의 참조 카운트가 최소값인 경우, 상기 메모리 박스에 할당된 메모리 영역을 해제하고, 상기 메모리 영역에 상응하는 매핑 테이블의 엔트리를 삭제할 수 있다.

[0032] 이 때, 상기 공유하는 단계는 상기 파라미터 공유 장치에 기록된 메모리 주소를 참조하여 상기 메모리 박스에 저장된 파라미터 값을 읽어오는(read) 단계; 모델 알고리즘을 이용하여 상기 메모리 박스의 파라미터 값에 상응하는 파라미터 차분 값을 계산하는 단계 및 상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 파라미터 값을 수정하는 단계를 포함할 수 있다.

[0033] 이 때, 상기 파라미터 값을 수정하는 단계는 상기 메모리 박스가 합저장 기능 수행이 가능한 경우, 상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 합저장 기능을 통해 상기 메모리 박스의 파라미터 값을 수정할 수 있다.

[0034] 이 때, 상기 파라미터 값을 수정하는 단계는 상기 메모리 박스가 합저장 기능 수행이 불가능한 경우, 상기 메모리 박스로부터 파라미터 값을 다시 읽어오고(read), 상기 파라미터 차분 값과 다시 읽어온 파라미터 값을 이용하여 산출한 파라미터 수정 값을 상기 메모리 박스에 쓸 수 있다(write).

[0035] 또한, 상기한 목적을 달성하기 위한 본 발명의 일실시예에 따른 파라미터 공유 장치는 메모리 박스에 파라미터가 저장될 메모리 영역의 할당 관리를 수행하고, 상기 메모리 영역의 할당 관리에 따라 상기 메모리 박스에 저장된 매핑 테이블을 업데이트 하는 메모리 할당부 및 상기 파라미터가 저장될 메모리 영역의 할당 관리를 위한 파라미터 정보를 상기 메모리 할당부에 제공하고, 상기 메모리 박스에 저장된 파라미터를 공유하는 연산 처리부를 포함한다.

[0036] 이 때, 상기 메모리 할당부는 상기 연산 처리부로부터 상기 메모리 영역의 할당 관리를 위한 상기 파라미터 정보를 수신하는 파라미터 정보 수신부; 상기 메모리 박스의 잠금을 관리하고, 상기 매핑 테이블을 업데이트하는 매핑 테이블 관리부 및 상기 매핑 테이블의 참조 카운트를 수정하여, 상기 메모리 영역의 할당 관리를 수행하는 공유 메모리 관리부를 포함할 수 있다.

[0037] 이 때, 상기 메모리 할당부는 상기 참조 카운트에 따라 상기 메모리 영역에서 파라미터를 공유 중인 다른 파라미터 공유 장치의 개수를 확인할 수 있다.

[0038] 이 때, 상기 메모리 박스는 상기 연산 처리부의 파라미터 값 수정 요청에 따라 합저장 기능을 이용하여 상기 메모리 박스에 저장된 파라미터 값을 업데이트할 수 있다.

발명의 효과

[0039] 본 발명은 분산 딥러닝 학습에서 다수의 워커 장치들이 파라미터를 공유하기 위해서 파라미터 서버를 사용하는 대신에, 공유 메모리 장치인 메모리 박스에서 제공하는 공유 메모리를 통해서 파라미터를 공유할 수 있다.

[0040] 또한, 본 발명은 통신 메시지 형태가 아니라 로컬 메모리 접근 방식으로 파라미터를 송수신 함으로써 통신 오버헤드 경감 및 메시지 처리 시간 감축을 통해서 딥러닝 학습을 가속화할 수 있다.

[0041] 또한, 본 발명은 분산 딥러닝 학습에서 파라미터 서버를 메모리 박스로 대체 지원함에 있어서 딥러닝 프레임워크가 가진 원래의 기능과 사용자가 사용하는 딥러닝 모델 개발 및 학습 인터페이스에 수정을 가하지 않고, 다수

의 워커 장치가 투명하게 메모리 박스를 통해 파라미터를 공유할 수 있다.

도면의 간단한 설명

[0042] 도 1은 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크를 나타낸 블록도이다.

도 2는 도 1에 도시된 메모리 박스의 일 예를 세부적으로 나타낸 블록도이다.

도 3은 본 발명의 일실시예에 따른 파라미터 공유 장치를 나타낸 블록도이다.

도 4는 도 3에 도시된 메모리 박스 접근부의 일 예를 세부적으로 나타낸 블록도이다.

도 5는 도 2 및 도 4에 도시된 메모리 박스 접근부와 메모리 박스의 일 예를 세부적으로 나타낸 블록도이다.

도 6은 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크에서 파라미터 공유를 나타낸 도면이다.

도 7은 본 발명의 일실시예에 따른 메모리 영역 할당을 위한 파라미터 공유 방법을 나타낸 동작흐름도이다.

도 8은 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법을 나타낸 동작흐름도이다.

도 9는 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 가능한 경우, 합저장 기능을 이용한 파라미터 공유 방법을 나타낸 시퀀스 다이어그램이다.

도 10은 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법을 나타낸 시퀀스 다이어그램이다.

발명을 실시하기 위한 구체적인 내용

[0043] 본 발명을 첨부된 도면을 참조하여 상세히 설명하면 다음과 같다. 여기서, 반복되는 설명, 본 발명의 요지를 불필요하게 흐릴 수 있는 공지 기능, 및 구성에 대한 상세한 설명은 생략한다. 본 발명의 실시형태는 당 업계에서 평균적인 지식을 가진 자에게 본 발명을 보다 완전하게 설명하기 위해서 제공되는 것이다. 따라서, 도면에서의 요소들의 형상 및 크기 등은 보다 명확한 설명을 위해 과장될 수 있다.

[0044] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성 요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0045] 이하, 본 발명에 따른 바람직한 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0046] 도 1은 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크를 나타낸 블록도이다. 도 2는 도 1에 도시된 메모리 박스의 일 예를 세부적으로 나타낸 블록도이다. 도 3은 본 발명의 일실시예에 따른 파라미터 공유 장치를 나타낸 블록도이다. 도 4는 도 3에 도시된 메모리 박스 접근부의 일 예를 세부적으로 나타낸 블록도이다.

[0047] 도 1을 참조하면, 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크는 복수개의 파라미터 공유 장치(10, 20, 30)들과 메모리 박스(1)로 구성된다.

[0048] 파라미터 공유 장치(10, 20, 30)들은 분산 딥러닝의 워커 장치라 불리는 독립적인 컴퓨터 장치에 상응할 수 있다.

[0049] 이 때, 파라미터 공유 장치(10, 20, 30)들은 메모리 박스(1)에 딥러닝 파라미터를 저장하고, 메모리 박스(1)에 저장된 딥러닝 파라미터를 공유하여 서로 협력적으로 학습할 수 있다.

[0050] 메모리 박스(1)는 전용 하드웨어로 구현된 공유 메모리 장치(device)에 상응할 수 있으며, 보다 빠르게 데이터를 저장하고 공유할 수 있다. 메모리 박스(1)는 다중 머신 간에 데이터를 저지연, 고속으로 공유 가능하게 하는 FPGA 통신 가속 공유 메모리 장치에 상응할 수 있다.

[0051] 메모리 박스(1)는 각 머신에 연결 가능한 PCIe 기반의 다수의 연결 커넥터를 보유하고 있어서, 각 머신에서 로컬 디바이스처럼 접근할 수 있으며 다중 머신에서 동시에 접근 할 수 있고, 대용량의 메모리를 가질 수 있다.

[0052] 또한, 메모리 박스(1)는 일반적인 네트워크보다 빠른 속도로 데이터를 리드(read)하거나 라이트(write)할 수 있다. 컴퓨터 노드에서는 DMA나 PIO 방식을 통해서 메모리 박스에 데이터를 읽거나 쓸 수 있다. 응용 프로그램은 메모리 박스의 디바이스 드라이버와 그 상위에 위치한 라이브러리를 통해서 메모리 박스(1)를 사용할 수 있다. 딥러닝 모델의 분산 병렬 학습에 있어서 메모리 박스(1)를 이용하면 워커 장치들 간에 파라미터를 저지연, 고속으로 공유할 수 있다.

[0053] 또한, 메모리 박스(1)는 데이터의 합저장(AssignAdd) 기능을 보유할 수도 있어서 딥러닝 파라미터를 효과적으로 업데이트 할 수 있다.

[0054] 도 2를 참조하면, 본 발명의 일실시예에 따른 메모리 박스(1)는 매핑 테이블 저장부(1a)와 파라미터 저장부(1b)를 포함한다.

[0055] 메모리 박스(1)는 매핑 테이블 저장부(1a)와 파라미터 저장부(1b)를 통해서 공유 메모리에 매핑 테이블과 딥러닝 파라미터를 저장할 수 있다.

[0056] 매핑 테이블은 각각의 엔트리에 파라미터의 식별자, 메모리 영역에 대한 메모리 주소 및 참조 카운트를 포함하는 매핑 정보가 기록될 수 있다.

[0057] 이 때, 매핑 테이블 저장부(1a)는 메모리 박스(1)와 파라미터를 공유하는 파라미터 공유 장치들(10, 20, 30)에 상응하는 각각의 매핑 정보를 기록할 수도 있다.

[0058] 도 3을 참조하면, 본 발명의 일실시예에 따른 파라미터 공유 장치 1(10)는 딥러닝 모델 복제부(11), 분산 딥러닝 학습 엔진부(12), CPU 장치 접근부(13), GPU 장치 접근부(14) 및 메모리 박스 접근부(100)를 포함할 수 있다.

[0059] 딥러닝 모델 복제부(11)는 파라미터를 학습할 수 있다.

[0060] 분산 딥러닝 학습 엔진부(12)는 딥러닝 모델 복제부(11)를 실행시키는 하부 엔진으로서, 메모리 박스(1)를 로컬에 위치한 독립적인 장치로 인식하여 메모리 박스 접근부(100)를 통해 메모리 박스(1)에 파라미터를 저장하거나, 메모리 박스(1)로부터 파라미터를 읽어와서 학습을 진행할 수 있다.

[0061] CPU 장치 접근부(13)는 CPU 에 접근할 수 있다.

[0062] GPU 장치 접근부(14)는 GPU 에 접근할 수 있다.

[0063] 메모리 박스 접근부(100)는 메모리 박스(1)에 접근할 수 있다.

[0064] CPU와 GPU는 계산 연산을 실행하는데 반해서 메모리 박스(1)는 계산 연산이 아니라 파라미터를 저장하기 위한 용도로 사용될 수 있다.

[0065] 메모리 박스 접근부(100)는 메모리 박스 장치 드라이버 또는 장치 드라이버의 상위에 위치한 메모리 박스 장치 라이브러리에서 제공하는 인터페이스를 통해서 메모리 박스(1)가 제공하는 기능을 이용할 수 있다.

[0066] 다시 말해서, 메모리 박스 접근부(100)는 메모리 박스(1)에서 제공하는 공유 메모리에 읽기, 쓰기, 잠금 걸기, 잠금 해제, 합저장 등의 기능을 이용할 수 있다.

[0067] CPU, GPU는 각 워커 장치들이 포함하는 구성이지만, 메모리 박스(1)는 여러 워커 장치들이 공유할 수 있다. 이 때, 메모리 박스 접근부(100)는 워커 장치들이 동일한 파라미터에 접근하는 경우, 동일한 메모리 주소에 접근하도록 하여 파라미터를 공유할 수 있다.

[0068] 이 때, 메모리 박스 접근부(100)는 모듈화되어 메모리 박스(1)와의 파라미터 공유를 위하여 기존의 워커 장치에 연결시켜 사용될 수 있다.

[0069] 즉, 메모리 박스 접근부(100)는 워커 장치에 연결시켜 메모리 박스(1)기반 분산형 딥러닝 파라미터를 공유하기 위한 파라미터 공유 장치 1(10)에 상응할 수도 있다.

[0070] 도 4를 참조하면, 본 발명의 일실시예에 따른 메모리 박스 접근부(100)는 메모리 할당부(110) 및 연산 처리부(120)를 포함할 수 있다.

[0071] 메모리 할당부(110)는 메모리 박스(1)에 파라미터가 저장될 메모리 영역의 할당 관리를 수행할 수 있고, 메모리 영역의 할당 관리에 따라 상기 메모리 박스에 저장된 매핑 테이블을 업데이트 할 수 있다.

[0072] 이 때, 메모리 할당부(113)는 참조 카운트에 따라 상기 메모리 영역에서 파라미터를 공유 중인 파라미터 공유 장치의 개수를 확인할 수 있고, 상기 참조 카운트가 최소값이 되는 경우, 상기 연산 처리부의 메모리 영역에 대한 메모리 주소를 삭제하고 상기 메모리 박스에서 상기 메모리 영역을 해제할 수 있다.

[0073] 연산 처리부(120)는 파라미터가 저장될 메모리 영역의 할당 관리를 위한 파라미터 정보를 메모리 할당부(110)에 제공하고, 메모리 박스(1)에 저장된 파라미터를 공유할 수 있다.

[0074] 이 때, 연산 처리부(120)는 메모리 박스(1)의 합저장 기능을 이용하여 메모리 박스(1)에 저장된 메모리 박스의 파라미터 값을 업데이트 할 수 있다.

[0075] 도 5는 도 2 및 도 4에 도시된 메모리 박스 접근부와 메모리 박스의 일 예를 세부적으로 나타낸 블록도이다.

[0076] 도 5를 참조하면, 메모리 할당부(110)는 파라미터 정보 수신부(111), 매핑 테이블 관리부(112) 및 공유 메모리 관리부(113)를 포함할 수 있다.

[0077] 연산 처리부(120)는 파라미터 객체부(121) 및 파라미터 처리부(122)를 포함할 수 있다.

[0078] 파라미터 정보 수신부(111)는 메모리 박스(1)에 파라미터를 저장할 메모리를 할당하거나 할당된 메모리를 해제하기 위해 필요한 파라미터에 관한 정보를 파라미터 객체부(121)로부터 수신할 수 있다.

[0079] 이 때, 파라미터 정보 수신부(111)는 파라미터의 파라미터 식별자, 파라미터 크기 및 파라미터가 저장된 메모리 영역에 대한 메모리 주소 중 적어도 하나를 포함하는 파라미터 정보를 파라미터 객체부(121)로부터 수신할 수 있다.

[0080] 매핑 테이블 관리부(112)는 메모리 박스(1)의 공유 메모리로부터 매핑 테이블을 읽어올 수 있다. 매핑 테이블은 파라미터와 파라미터가 저장된 공유 메모리 주소의 매핑 정보를 관리하는 테이블에 상응할 수 있다. 매핑 테이블의 각 엔트리에는 파라미터 식별자, 공유 메모리 주소, 그리고 참조 카운트를 포함할 수 있다.

[0081] 공유 메모리 관리부(113)는 메모리 박스(1)에 파라미터의 메모리 영역을 할당할 수 있다.

[0082] 공유 메모리 관리부(113)는 매핑 테이블 관리부(112)에서 읽어온 매핑 테이블을 검색하여 메모리 박스(1)에 상기 파라미터를 저장할 메모리 영역의 할당 여부를 판단할 수 있다.

[0083] 이 때, 공유 메모리 관리부(113)는 매핑 테이블에서 파라미터를 검색하여 파라미터의 메모리 영역이 할당 되었는지 여부를 확인하고, 메모리가 할당되어 있는 경우, 매핑 테이블의 엔트리에 참조 카운트를 증가시키고 메모리 주소를 파라미터 객체부(121)에 기록할 수 있다.

[0084] 이 때, 공유 메모리 관리부(113)는 메모리가 아직 할당되지 않은 경우, 메모리 박스(1)에 상기 파라미터를 저장할 메모리 영역을 할당하고 매핑 테이블에 새로운 엔트리를 추가하며, 할당한 메모리 주소를 파라미터 객체부(121)에 기록할 수 있다. 매핑 테이블 관리부(112)는 수정된 매핑 테이블을 메모리 박스(1)에 쓸 수 있다(write). 이 때, 매핑 테이블 관리부(112)는 메모리 박스(1)의 메모리 영역에 대해 잠금을 걸거나 잠금을 해제할 수 있다.

[0085] 이 때, 매핑 테이블 관리부(112)는 매핑 테이블을 읽어오기 전에 메모리 박스(1)의 매핑 테이블이 저장된 메모리 영역에 잠금을 걸고, 메모리 박스(1)에 수정된 매핑 테이블을 쓴 후에 상기 잠금을 해제할 수 있다.

[0086] 메모리 주소는 메모리 박스(1)의 메모리 영역 전체에서 특정한 메모리 위치를 지정한 것에 상응할 수 있다. 메모리 주소는 메모리 박스의 디바이스 드라이버 및 접근 라이브러리에서 제공하는 방식에 따르며, 디바이스 메모리 주소 또는 이와 매핑된 가상 주소 또는 디바이스 메모리 주소에 매핑된 식별자 등에 상응할 수 있다.

[0087] 또한, 공유 메모리 관리부(113)는 메모리 박스(1)에 할당된 파라미터의 메모리 영역을 해제할 수 있다.

[0088] 공유 메모리 관리부(113)는 매핑 테이블 관리부(112)가 읽어온 매핑 테이블을 검색하여 메모리 박스(1)에서 파라미터가 할당된 메모리 영역의 해제 여부를 판단할 수 있다.

[0089] 이 때, 공유 메모리 관리부(113)는 매핑 테이블에서 파라미터 식별자 또는 메모리 주소를 이용하여 메모리 영역을 해제할 파라미터에 관한 엔트리를 검색한 후, 해당 엔트리의 참조 카운트를 감소시킬 수 있다.

[0090] 이 때, 공유 메모리 관리부(113)는 참조 카운트값에 따라서 메모리 영역의 해제 여부를 판단할 수 있다. 이 때, 공유 메모리 관리부(113)는 참조 카운트가 최소값(예를 들어, 0)이면 메모리 영역을 해제할 수 있고, 참조 카운트가 최소값이 아닌 경우, 메모리 영역의 해제를 수행하지 않을 수 있다.

[0091] 이 때, 공유 메모리 관리부(113)는 메모리 영역을 해제하도록 결정한 경우, 메모리 박스(1)에서 파라미터의 메모리 영역을 해제하고, 매핑 테이블에서 해당 엔트리를 삭제할 수 있다. 매핑 테이블 관리부(112)는 수정된 매핑 테이블을 메모리 박스에 쓸 수 있다(write). 매핑 테이블 관리부(112)는 메모리 박스(1)의 메모리 영역에 대해 잠금을 걸거나 잠금을 해제할 수 있다. 매핑 테이블 관리부(112)는 매핑 테이블을 읽어오기 전에 메모리 박스(1)의 매핑 테이블이 저장된 메모리 영역에 잠금을 걸고, 메모리 박스(1)에 매핑 테이블을 쓴 후에 잠금을 해

제할 수 있다.

[0092] 파라미터 객체부(121)는 파라미터의 메모리 영역에 대한 메모리 주소를 파라미터 정보 수신부(111)에 파라미터 정보로 제공할 수 있다.

[0093] 파라미터 처리부(122)는 파라미터 객체(121)에 기록된 메모리 주소를 통해서 메모리 박스(1)에 파라미터 값을 쓰기(write), 읽기(read) 및 메모리 박스(1)에서 제공하는 합저장(AssignAdd) 기능을 이용하여 학습한 파라미터를 수정(업데이트)할 수 있다.

[0094] 이 때, 파라미터 처리부(122)는 메모리 박스(1)에 저장된 메모리 박스의 파라미터 값을 읽고(read), 모델 알고리즘을 이용하여 상기 메모리 박스의 파라미터 값에 상응하는 파라미터 차분 값을 계산할 수 있다.

[0095] 이 때, 모델 알고리즘은 확률적 경사 하강 법(Stochastic Gradient Descent) 알고리즘이 사용될 수 있으며, 이외에도 파라미터 차분 값을 계산하기 위한 다양한 알고리즘이 사용될 수 있다.

[0096] 이 때, 파라미터 처리부(122)는 메모리 박스(1)의 합저장 기능 수행 가능 여부에 따라서, 메모리 박스(1)가 합저장 기능 수행이 가능한 경우, 상기 합저장 기능을 통해 상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 파라미터 값을 수정할 수 있다.

[0097] 이 때, 파라미터 처리부(122)는 메모리 박스(1)의 합저장 기능 수행 가능 여부에 따라서, 메모리 박스(1)가 합저장 기능 수행이 불가능한 경우, 상기 파라미터 차분 값과 상기 메모리 박스의 파라미터 값에 대한 파라미터 수정 값을 산출하여 상기 메모리 박스에 쓸 수 있다(write).

[0098] 도 6은 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크에서 파라미터 공유를 나타낸 도면이다.

[0099] 도 6을 참조하면, 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크에서 파라미터 공유 기법은 파라미터 공유 장치들(10, 20, 30)은 메모리 박스 접근부(100, 200, 300)을 통해 각자 독립적으로 복수개의 파라미터들을 학습할 수 있다.

[0100] 이 때, 파라미터 공유 장치들(10, 20, 30)은 메모리 박스 접근부(100, 200, 300)를 통해서 메모리 박스(1)에 복수개의 파라미터들을 저장하고 학습을 수행할 수 있다. 메모리 박스 접근부(100, 200, 300)는 파라미터 공유 장치들(10, 20, 30)이 메모리 박스(1)의 동일한 파라미터에 접근하는 경우에는 동일한 메모리 주소에 접근하여 동일한 파라미터를 공유할 수 있다.

[0101] 이 때, 파라미터 공유 장치들(10, 20, 30)은 동일한 메모리 주소에 파라미터를 업데이트하고, 업데이트한 파라미터를 읽어갈 수 있다. 따라서, 파라미터 공유 장치들(10, 20, 30)는 복수개의 파라미터들을 서로 협력적으로 학습할 수 있다.

[0102] 본 발명의 일실시예에 따른 파라미터 공유 방법은 메모리 영역 할당을 위한 파라미터 공유 방법과 할당된 메모리 영역의 해제를 위한 파라미터 공유 방법을 나눠서 설명한다.

[0103] 도 7은 본 발명의 일실시예에 따른 메모리 영역 할당을 위한 파라미터 공유 방법을 나타낸 동작흐름도이다.

[0104] 도 7을 참조하면, 본 발명의 일실시예에 따른 메모리 영역 할당을 위한 파라미터 공유 방법은 먼저 파라미터 정보를 수신할 수 있다(S210).

[0105] 즉, 단계(S210)는 메모리 박스(1)에 파라미터의 메모리 영역을 할당하기 위한 파라미터 정보를 수신할 수 있다.

[0106] 이 때, 단계(S210)는 파라미터의 파라미터 식별자 및 파라미터 크기 중 적어도 하나를 포함하는 파라미터 정보를 수신할 수 있다.

[0107] 이 때, 파라미터 크기는 파라미터를 저장하기 위해 필요한 메모리 크기일 수 있다.

[0108] 이 때, 매핑 테이블은 각각의 엔트리에 파라미터의 식별자, 메모리 영역에 대한 메모리 주소 및 참조 카운트를 포함하는 매핑 정보가 기록될 수 있다.

[0109] 이 때, 매핑 테이블은 메모리 박스(1)와 파라미터를 공유하는 파라미터 공유 장치들(10, 20, 30)에 상응하는 각각의 매핑 정보가 기록될 수도 있다.

[0110] 즉, 매핑 테이블은 메모리 박스(1)와 파라미터를 공유하는 파라미터 공유 장치들(10, 20, 30)에 관한 정보를 사전에 더 포함할 수도 있다.

[0111] 또한, 본 발명의 일실시예에 따른 메모리 영역 할당을 위한 파라미터 공유 방법은 매핑 테이블의 잠금 및 읽기를 수행할 수 있다(S220).

[0112] 즉, 단계(S220)는 메모리 박스(1)의 매핑 테이블에 잠금을 걸고, 매핑 테이블을 읽어올 수 있다.

[0113] 또한, 본 발명의 일실시예에 따른 메모리 영역 할당을 위한 파라미터 공유 방법은 메모리 영역의 할당 여부를 확인할 수 있다(S230).

[0114] 즉, 단계(S230)는 매핑 테이블의 엔트리를 확인하여 메모리 박스에 파라미터의 메모리 영역이 할당되지 않은 경우, 메모리 영역을 할당할 수 있다(S240).

[0115] 이 때, 단계(S240)는 파라미터 크기만큼 상기 메모리 박스에 메모리 영역을 할당할 수 있다.

[0116] 또한, 본 발명의 일실시예에 따른 메모리 영역 할당을 위한 파라미터 공유 방법은 매핑 테이블에 매핑 정보를 추가할 수 있다(S250).

[0117] 즉, 단계(S250)는 매핑 테이블의 엔트리를 확인하여 메모리 박스에 파라미터의 메모리 영역이 할당되지 않은 경우, 메모리 영역이 할당된 파라미터에 대한 매핑 정보를 매핑 테이블의 새로운 엔트리에 추가하여 매핑 테이블을 업데이트 할 수 있다.

[0118] 또한, 단계(S230)는 매핑 테이블의 엔트리를 확인하여 메모리 박스에 파라미터의 메모리 영역이 기 할당되어 있는 경우, 매핑 테이블의 참조 카운트를 증가시킬 수 있다(S260).

[0119] 즉, 단계(S260)는 매핑 테이블에 파라미터에 상응하는 엔트리에 참조 카운트를 증가시켜 매핑 테이블을 업데이트 할 수 있다.

[0120] 이 때, 단계(S260)는 참조 카운트를 '1'씩 증가 시킬 수 있다.

[0121] 또한, 본 발명의 일실시예에 따른 메모리 영역 할당을 위한 파라미터 공유 방법은 매핑 테이블의 쓰기 및 잠금 해제를 수행할 수 있다(S270).

[0122] 즉, 단계(S270)는 메모리 영역의 할당 여부에 따라 매핑 정보를 수정한 매핑 테이블을 메모리 박스(1)에 쓰고, 매핑 테이블의 잠금을 해제할 수 있다.

[0123] 이 때, 단계(S270)는 메모리 영역이 할당된 파라미터에 대한 메모리 박스의 메모리 주소를 파라미터 공유 장치에 기록할 수 있다.

[0124] 또한, 본 발명의 일실시예에 따른 메모리 영역 할당을 위한 파라미터 공유 방법은 파라미터를 공유할 수 있다(S280).

[0125] 즉, 단계(S280)는 메모리 영역을 할당한 메모리 주소를 고려하여 파라미터를 공유할 수 있다.

[0126] 이 때, 단계(S280)는 파라미터 공유 장치들(10, 20, 30)에 메모리 영역을 할당한 메모리 주소가 추가된 것으로 기록된 메모리 영역의 메모리 주소를 참조하여 메모리 박스(1)에 저장된 파라미터를 공유할 수 있다.

[0127] 이 때, 단계(S280)는 메모리 박스(1)의 매핑 테이블에 기록된 파라미터 공유 장치들(10, 20, 30)에 상응하는 메모리 주소를 참조하여 메모리 박스(1)에 저장된 파라미터를 공유할 수도 있다.

[0128] 이 때, 단계(S280)는 파라미터 공유 장치들(10, 20, 30)이 메모리 박스(1)에 저장된 파라미터 값을 읽어 올 수 있다(read).

[0129] 이 때, 단계(S280)는 모델 알고리즘을 이용하여 메모리 박스(1)의 파라미터 값에 상응하는 파라미터 차분 값을 계산할 수 있다.

[0130] 이 때, 모델 알고리즘은 확률적 경사 하강 법(Stochastic Gradient Descent) 알고리즘이 사용될 수 있으며, 이외에도 파라미터 차분 값을 계산하기 위한 다양한 알고리즘이 사용될 수 있다.

[0131] 이 때, 단계(S280)는 파라미터 차분 값을 이용하여 메모리 박스(1)의 파라미터 값을 수정할 수 있다.

[0132] 이 때, 단계(S280)는 메모리 박스(1)가 합저장 기능 수행이 가능한 경우, 메모리 박스(1)의 합저장 기능을 통해 상기 파라미터 차분 값을 이용하여 상기 메모리 박스(1)의 파라미터 값을 수정할 수 있다.

[0133] 이 때, 단계(S280)는 파라미터 공유 장치들(10, 20, 30)이 메모리 박스(1)의 합저장 기능 수행 가능 여부를 미

리 확인해둘 수 있다.

[0134] 또한, 단계(S280)는 메모리 박스(1)가 합저장 기능 수행이 불가능한 경우, 메모리 박스(1)로부터 파라미터 값을 다시 읽어오고(read), 파라미터 차분 값과 다시 읽어온 파라미터 값을 이용하여 산출한 파라미터 수정 값을 메모리 박스(1)에 쓸 수 있다(write).

[0135] 이 때, 단계(S280)는 한 번 또는 그 이상 반복적으로 실행될 수 있다.

[0136] 즉, 단계(S280)는 메모리 박스(1)가 공유하는 파라미터를 이용하여 학습을 진행할 수 있다.

[0137] 이 때, 단계(S280)는 파라미터 공유 장치들(10, 20, 30)이 동일한 파라미터를 접근하는 경우에는 동일한 메모리 주소에 접근하도록 하여 동일한 파라미터가 공유될 수 있다.

[0138] 이 때, 단계(S280)는 파라미터 공유 장치들(10, 20, 30)이 동일한 메모리 주소에서 파라미터를 읽어가고, 업데이트를 하고, 업데이트된 파라미터를 다시 읽어가서 파라미터 공유 장치들(10, 20, 30)이 서로 협력적으로 학습할 수 있다.

[0139] 나아가, 단계(S280)에서 파라미터를 공유하는 과정은 도 9 및 도 10에 대한 설명을 일 예로 하여 아래에서 상세하게 설명한다.

[0140] 도 8은 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법을 나타낸 동작흐름도이다.

[0141] 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법은 먼저 파라미터 정보를 수신할 수 있다(S310).

[0142] 즉, 단계(S310)는 메모리 박스(1)에 파라미터의 메모리 영역을 해제하기 위한 파라미터 정보를 수신할 수 있다.

[0143] 이 때, 단계(S310)는 파라미터의 파라미터 식별자 및 파라미터가 저장된 메모리 영역에 대한 메모리 주소 중 적어도 하나를 포함하는 파라미터 정보를 수신할 수 있다.

[0144] 이 때, 매핑 테이블은 각각의 엔트리에 파라미터의 식별자, 메모리 영역에 대한 메모리 주소 및 참조 카운트를 포함하는 매핑 정보가 기록될 수 있다.

[0145] 이 때, 매핑 테이블은 메모리 박스(1)와 파라미터를 공유하는 파라미터 공유 장치들(10, 20, 30)에 상응하는 각각의 매핑 정보가 기록될 수도 있다.

[0146] 즉, 매핑 테이블은 메모리 박스(1)와 파라미터를 공유하는 파라미터 공유 장치들(10, 20, 30)에 관한 정보를 사전에 더 포함할 수도 있다.

[0147] 또한, 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법은 매핑 테이블의 잠금 및 읽기를 수행할 수 있다(S320).

[0148] 즉, 단계(S320)는 메모리 박스(1)의 매핑 테이블에 잠금을 걸고, 매핑 테이블을 읽어올 수 있다.

[0149] 또한, 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법은 참조 카운트를 감소시킬 수 있다(S330).

[0150] 즉, 단계(S330)는 매핑 테이블에 파라미터에 상응하는 엔트리에 참조 카운트를 감소시켜 매핑 테이블을 업데이트 할 수 있다.

[0151] 이 때, 단계(S330)는 참조 카운트를 '1'씩 감소 시킬 수 있다.

[0152] 또한, 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법은 메모리 주소를 삭제할 수 있다(S340).

[0153] 즉, 단계(S340)는 파라미터 공유 장치에 기록된 파라미터에 상응하는 메모리 주소를 삭제할 수 있다.

[0154] 또한, 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법은 참조 카운트가 최소값(예를 들어, '0')인지 여부를 확인할 수 있다(S350).

[0155] 즉, 단계(S350)는 매핑 테이블의 참조 카운트가 최소값인 경우, 메모리 박스(1)에 할당된 메모리 영역을 해제하고(S360), 매핑 테이블의 매핑 정보를 삭제할 수 있다(S370).

[0156] 즉, 단계(S370)는 메모리 영역에 상응하는 매핑 테이블의 엔트리를 삭제할 수 있다.

[0157] 또한, 단계(S350)는 매핑 테이블의 참조 카운트가 최소값이 아닌 경우, 메모리 영역을 해제 하지 않고, 참조 카운트가 수정된 매핑 테이블을 업데이트 할 수 있다.

[0158] 또한, 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법은 매핑 테이블 쓰기 및 잠금 해제를 수행할 수 있다(S380).

[0159] 즉, 단계(S380)는 메모리 영역의 해제 여부에 따라 매핑 정보를 수정한 매핑 테이블을 메모리 박스(1)에 쓰고, 매핑 테이블의 잠금을 해제할 수 있다.

[0160] 또한, 본 발명의 일실시예에 따른 메모리 영역 해제를 위한 파라미터 공유 방법은 파라미터를 공유할 수 있다(S390).

[0161] 즉, 단계(S390)는 메모리 영역을 해제한 메모리 주소를 고려하여 파라미터를 공유할 수 있다.

[0162] 이 때, 단계(S390)는 파라미터 공유 장치들(10, 20, 30)에 메모리 영역을 해제한 메모리 주소가 삭제된 것으로 기록된 나머지 메모리 영역에 대한 메모리 주소를 참조하여 메모리 박스(1)에 저장된 파라미터를 공유할 수 있다.

[0163] 이 때, 단계(S390)는 메모리 박스(1)의 매핑 테이블에 기록된 파라미터 공유 장치들(10, 20, 30)에 상응하는 메모리 주소를 참조하여 메모리 박스(1)에 저장된 파라미터를 공유할 수도 있다.

[0164] 이 때, 단계(S390)는 파라미터 공유 장치에 기록된 메모리 주소를 참조하여 메모리 박스(1)에 저장된 파라미터 값을 읽어 올 수 있다(read).

[0165] 이 때, 단계(S390)는 모델 알고리즘을 이용하여 메모리 박스의 파라미터 값에 상응하는 파라미터 차분 값을 계산할 수 있다.

[0166] 이 때, 모델 알고리즘은 확률적 경사 하강 법(Stochastic Gradient Descent) 알고리즘이 사용될 수 있으며, 이외에도 파라미터 차분 값을 계산하기 위한 다양한 알고리즘이 사용될 수 있다.

[0167] 이 때, 단계(S390)는 파라미터 차분 값을 이용하여 상기 메모리 박스의 파라미터 값을 수정할 수 있다.

[0168] 이 때, 단계(S390)는 상기 메모리 박스가 합저장 기능 수행이 가능한 경우, 상기 파라미터 차분 값을 이용하여 상기 메모리 박스의 합저장 기능을 통해 상기 메모리 박스의 파라미터 값을 수정할 수 있다.

[0169] 이 때, 단계(S390)는 파라미터 공유 장치들(10, 20, 30)이 메모리 박스(1)의 합저장 기능 수행 가능 여부를 미리 확인해둘 수 있다.

[0170] 이 때, 단계(S390)는 상기 메모리 박스가 합저장 기능 수행이 불가능한 경우, 상기 메모리 박스로부터 파라미터 값을 다시 읽어오고(read), 상기 파라미터 차분 값과 다시 읽어온 파라미터 값을 이용하여 산출한 파라미터 수정 값을 상기 메모리 박스에 쓸 수 있다(write).

[0171] 이 때, 단계(S390)는 한 번 또는 그 이상 반복적으로 실행될 수 있다.

[0172] 즉, 단계(S390)는 메모리 박스(1)가 공유하는 파라미터를 이용하여 학습을 진행할 수 있다.

[0173] 이 때, 단계(S390)는 파라미터 공유 장치들(10, 20, 30)이 동일한 파라미터를 접근하는 경우에는 동일한 메모리 주소에 접근하도록 하여 동일한 파라미터가 공유될 수 있다.

[0174] 이 때, 단계(S390)는 파라미터 공유 장치들(10, 20, 30)이 동일한 메모리 주소에서 파라미터를 읽어가고, 업데이트를 하고, 업데이트된 파라미터를 다시 읽어가서 파라미터 공유 장치들(10, 20, 30)이 서로 협력적으로 학습할 수 있다.

[0175] 나아가, 단계(S390)에서 파라미터를 공유하는 과정은 도 9 및 도 10에 대한 설명을 일 예로 하여 아래에서 상세하게 설명한다.

[0176] 도 9는 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 가능한 경우, 합저장 기능을 이용한 파라미터 공유 방법을 나타낸 시퀀스 다이어그램이다.

[0177] 도 9를 참조하면, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 가능한 경우, 합저장 기능을 이용한 파라미터 공유 방법은 먼저 파라미터 공유 장치 1(10)이 파라미터를 읽어올 수 있다(S410).

[0178] 즉, 단계(S410)는 파라미터 공유 장치 1(10)이 메모리 박스(1)로부터 제1 메모리 박스의 파라미터 값을 읽어올

수 있다.

[0179] 또한, 본 발명의 일실시예에 따른 합저장 기능을 이용한 파라미터 공유 방법은 파라미터 차분 값을 계산할 수 있다(S420).

[0180] 즉, 단계(S420)는 파라미터 공유 장치 1(10)이 모델 알고리즘을 이용하여 제1 파라미터 차분 값을 계산할 수 있다.

[0181] 이 때, 모델 알고리즘은 확률적 경사 하강 법(Stochastic Gradient Descent) 알고리즘이 사용될 수 있으며, 이 외에도 파라미터 차분 값을 계산하기 위한 다양한 알고리즘이 사용될 수 있다.

[0182] 또한, 본 발명의 일실시예에 따른 합저장 기능을 이용한 파라미터 공유 방법은 합저장 기능을 이용하여 파라미터를 수정할 수 있다(S430).

[0183] 즉, 단계(S430)는 메모리 박스(1)의 합저장(AssignAdd) 기능을 통해 제1 파라미터 차분 값을 이용하여 제1 메모리 박스의 파라미터 값을 수정(업데이트)하여 제2 메모리 박스의 파라미터 값을 생성할 수 있다.

수학식 1

[0184] $$W_{t+1} = W_t + \triangle W_t$$

[0185] 예를 들어, 수학식 1은 합저장 기능의 일 예를 수학식으로 나타낸 것을 알 수 있다.

[0186] 이 때, 단계(S430)는 수학식 1과 같이, 제1 메모리 박스의 파라미터 값(W_t)에서 제1 파라미터 차분값(ΔW_t)을 합산하여 제2 메모리 박스의 파라미터 값(W_{t+1})을 생성할 수 있다.

[0187] 또한, 본 발명의 일실시예에 따른 합저장 기능을 이용한 파라미터 공유 방법은 파라미터 공유 장치 2(20)가 파라미터를 읽어올 수 있다(S440).

[0188] 즉, 단계(S440)는 파라미터 공유 장치 2(20)가 메모리 박스(1)로부터 제2 메모리 박스의 파라미터 값을 읽어올 수 있다.

[0189] 이 때, 단계(S440)는 파라미터 공유 장치 1(10)과 파라미터 공유 장치(2)가 비동기적으로 파라미터 공유를 수행하게 되므로, 도 9에 도시된 바와 같이 반드시 단계(S430) 이후에 수행되는 것이 아니라, 파라미터 공유 장치(1)의 파라미터 업데이트 과정과 무관하게 단계(S410) 내지 단계(S430) 중 어느 단계에서도 함께 수행될 수도 있다.

[0190] 따라서, 단계(S440)는 합저장 기능을 통해 제1 파라미터 차분 값이 업데이트 되지 않은 제1 메모리 박스의 파라미터 값을 읽어 올 수도 있다.

[0191] 그러나, 이하에서는 업데이트가 완료된 제2 메모리 박스의 파라미터 값을 읽어 오는 것으로 설명한다.

[0192] 이러한 비동기적 파라미터 공유 방법에서, 파라미터 값의 업데이트 과정 중에는 메모리 박스의 파라미터 값의 업데이트가 정확한 계산 값으로 반영되지 않을 수 있지만, 파라미터 공유가 완료되는 시점에서는 결과적으로 높은 속도로 파라미터 공유를 완료할 수 있다.

[0193] 또한, 본 발명의 일실시예에 따른 합저장 기능을 이용한 파라미터 공유 방법은 파라미터 차분 값을 계산할 수 있다(S450).

[0194] 즉, 단계(S450)는 파라미터 공유 장치 2(20)가 모델 알고리즘을 이용하여 제2 파라미터 차분 값을 계산할 수 있다.

[0195] 이 때, 모델 알고리즘은 확률적 경사 하강 법(Stochastic Gradient Descent) 알고리즘이 사용될 수 있으며, 이 외에도 파라미터 차분 값을 계산하기 위한 다양한 알고리즘이 사용될 수 있다.

[0196] 또한, 본 발명의 일실시예에 따른 합저장 기능을 이용한 파라미터 공유 방법은 합저장 기능을 이용하여 파라미터를 수정할 수 있다(S460).

[0197] 즉, 단계(S460)는 메모리 박스(1)의 합저장(AssignAdd) 기능을 통해 제2 파라미터 차분 값을 이용하여 제2 메모

리 박스의 파라미터 값을 수정(업데이트)하여 제3 메모리 박스의 파라미터 값을 생성할 수 있다.

[0198] 이 때, 단계(S460)는 수학식 1과 같이, 제2 메모리 박스의 파라미터 값(W_t)에서 제2 파라미터 차분값(ΔW_t)을 합산하여 제3 메모리 박스의 파라미터 값(W_{t+1})을 생성할 수 있다.

[0199] 이러한 과정을 통해, 복수개의 파라미터 공유 장치들(워커)이 비동기적으로 메모리 박스(1)로부터 파라미터 값을 읽어(read) 나가면서 합저장(AssignAdd) 기능을 이용하여 메모리 박스(1)의 파라미터 값을 업데이트 할 수 있다.

[0200] 도 10는 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법을 나타낸 시퀀스 다이어그램이다.

[0201] 도 10을 참조하면, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 먼저 파라미터 공유 장치 1(10)이 파라미터를 읽어올 수 있다(S510).

[0202] 즉, 단계(S510)는 파라미터 공유 장치 1(10)이 메모리 박스(1)로부터 제1 메모리 박스의 파라미터 값을 읽어올 수 있다(read).

[0203] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 차분 값을 계산할 수 있다(S520).

[0204] 즉, 단계(S520)는 파라미터 공유 장치 1(10)이 모델 알고리즘을 이용하여 제1 파라미터 차분 값을 계산할 수 있다.

[0205] 이 때, 모델 알고리즘은 확률적 경사 하강 법(Stochastic Gradient Descent) 알고리즘이 사용될 수 있으며, 이외에도 파라미터 차분 값을 계산하기 위한 다양한 알고리즘이 사용될 수 있다.

[0206] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 값을 읽어올 수 있다(S530).

[0207] 즉, 단계(S530)는 단계(S520)의 파라미터 차분 값 계산 과정 동안 다른 파라미터 공유 장치에 의하여 메모리 박스(1)의 파라미터 값이 수정될 수 있으므로, 메모리 박스(1)의 파라미터 값을 다시 읽어올 수 있다.

[0208] 이 때, 단계(S530)는 단계(S520)의 파라미터 차분 값의 계산이 기설정된 시간을 초과할 때까지 계산되지 않은 경우에만, 제1 메모리 박스의 파라미터 값을 다시 읽어 올 수 있다.

[0209] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 수정 값을 산출할 수 있다(S540).

[0210] 즉, 단계(S540)는 계산된 제1 파라미터 차분 값과 메모리 박스(1)에서 읽어온 제1 메모리 박스의 파라미터 값을 이용하여 제1 파라미터 수정 값을 산출 할 수 있다.

[0211] 이 때, 단계(S540)는 단계(S520)에서 기설정된 시간 이내에 파라미터 차분 값이 계산된 경우, 단계(S510)에서 읽어온 제1 메모리 박스의 파라미터 값을 이용하여 제1 파라미터 수정 값을 산출할 수 있다.

[0212] 또한, 단계(S540)는 단계(S520)에서 기설정된 시간을 초과하여 파라미터 차분 값이 계산된 경우, 단계(S530)에서 다시 읽어온 제1 메모리 박스의 파라미터 값을 이용하여 제1 파라미터 수정 값을 산출할 수 있다.

수학식 2

[0213] $$W_{t+1} = W_t' + \triangle W_t$$

[0214] 예를 들어, 수학식 2는 파라미터 값을 업데이트하는 일 예를 수학식으로 나타낸 것을 알 수 있다.

[0215] 이 때, 단계(S540)는 상기 수학식 2와 같이, 제1 메모리 박스의 파라미터 값(W_t')에서 제1 파라미터 차분값(ΔW_t)을 합산하여 제1 파라미터 수정 값(W_{t+1})을 생성할 수 있다.

[0216] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 수정 값을 쓸 수 있다(S550).

[0217] 즉, 단계(S550)는 산출된 제1 파라미터 수정 값을 메모리 박스(1)에 쓰는(write) 것으로 메모리 박스(1)의 파라미터 값을 수정(업데이트)할 수 있다.

[0218] 이 때, 단계(S550)는 제1 메모리 박스의 파라미터 값에 제1 파라미터 수정 값을 쓰는 것으로, 제2 메모리 박스의 파라미터 값을 생성할 수 있다.

[0219] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 공유 장치 2(20)가 파라미터를 읽어올 수 있다(S560).

[0220] 즉, 단계(S560)는 파라미터 공유 장치 2(10)가 메모리 박스(1)로부터 제2 메모리 박스의 파라미터 값을 읽어올 수 있다(read).

[0221] 이 때, 단계(S560)는 파라미터 공유 장치 1(10)과 파라미터 공유 장치(2)가 비동기적으로 파라미터 공유를 수행하게 되므로, 도 10에 도시된 바와 같이 반드시 단계(S530) 이후에 수행되는 것이 아니라, 파라미터 공유 장치(1)의 파라미터 업데이트 과정과 무관하게 단계(S510) 내지 단계(S550) 중 어느 단계에서도 수행될 수도 있다.

[0222] 따라서, 단계(S560)는 도 10에 도시된 바와 같이 제1 파라미터 수정 값이 업데이트 되지 않은 제1 메모리 박스의 파라미터 값을 읽어 올 수도 있다.

[0223] 그러나, 이하에서는 업데이트가 완료된 제2 메모리 박스의 파라미터 값을 읽어 오는 것으로 설명한다.

[0224] 이러한 비동기적 파라미터 공유 방법에서, 파라미터 값의 업데이트 과정 중에는 메모리 박스의 파라미터 값의 업데이트가 정확한 계산값으로 반영되지 않을 수 있지만, 파라미터 공유가 완료되는 시점에서는 결과적으로 높은 속도로 파라미터 공유를 완료할 수 있다.

[0225] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 차분 값을 계산할 수 있다(S570).

[0226] 즉, 단계(S570)는 파라미터 공유 장치 2(20)가 모델 알고리즘을 이용하여 제2 파라미터 차분 값을 계산할 수 있다.

[0227] 이 때, 모델 알고리즘은 확률적 경사 하강 법(Stochastic Gradient Descent) 알고리즘이 사용될 수 있으며, 이외에도 파라미터 차분 값을 계산하기 위한 다양한 알고리즘이 사용될 수 있다.

[0228] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 값을 읽어올 수 있다(S580).

[0229] 즉, 단계(S580)는 단계(S570)의 파라미터 차분 값 계산 과정 동안 다른 파라미터 공유 장치에 의하여 메모리 박스(1)의 파라미터 값이 수정될 수 있으므로, 메모리 박스(1)의 파라미터 값을 다시 읽어올 수 있다.

[0230] 이 때, 단계(S580)는 단계(S570)의 파라미터 차분 값이 계산이 기설정된 시간을 초과할 때까지 계산되지 않은 경우에만, 제2 메모리 박스의 파라미터 값을 다시 읽어 올 수 있다.

[0231] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 수정 값을 산출할 수 있다(S590).

[0232] 즉, 단계(S590)는 계산된 제2 파라미터 차분 값과 메모리 박스(1)에서 읽어온 제2 메모리 박스의 파라미터 값을 이용하여 제2 파라미터 수정 값을 산출 할 수 있다.

[0233] 이 때, 단계(S590)는 단계(S570)에서 기설정된 시간 이내에 파라미터 차분 값이 계산된 경우, 단계(S560)에서 읽어온 제2 메모리 박스의 파라미터 값을 이용하여 제2 파라미터 수정 값을 산출할 수 있다.

[0234] 또한, 단계(S590)는 단계(S570)에서 기설정된 시간을 초과하여 파라미터 차분 값이 계산된 경우, 단계(S580)에서 다시 읽어온 제2 메모리 박스의 파라미터 값을 이용하여 제2 파라미터 수정 값을 산출할 수 있다.

[0235] 이 때, 단계(S590)는 상기 수학식 2와 같이, 제2 메모리 박스의 파라미터 값(W_t')에서 제2 파라미터 차분값(ΔW_t)을 합산하여 제2 파라미터 수정 값(W_{t+1})을 생성할 수 있다.

[0236] 또한, 본 발명의 일실시예에 따른 메모리 박스가 합저장 기능 수행이 불가능한 경우, 파라미터 공유 방법은 파라미터 수정 값을 쓸 수 있다(S600).

[0237] 즉, 단계(S600)는 산출된 제2 파라미터 값을 메모리 박스(1)에 쓰는(write) 것으로 메모리 박스(1)의 파라미터

값을 수정(업데이트)할 수 있다.

[0238] 이 때, 단계(S600)는 제2 메모리 박스의 파라미터 값에 제2 파라미터 수정 값을 쓰는 것으로, 제3 메모리 박스의 파라미터 값을 생성할 수 있다.

[0239] 이러한 과정을 통해, 메모리 박스(1)가 합저장 기능 수행이 불가능한 경우에도, 복수개의 파라미터 공유 장치들(워커)이 비동기적으로 메모리 박스(1)로부터 파라미터 값을 읽어(read) 나가면서 산출된 파라미터 수정 값을 쓰는(write) 것으로, 메모리 박스(1)의 파라미터 값을 업데이트 할 수 있다.

[0240] 이상에서와 같이 본 발명에 따른 파라미터 공유 장치 및 방법은 상기한 바와 같이 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상기 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

부호의 설명

[0241] 1: 메모리 박스 1a: 매핑 테이블 저장부

1b: 파라미터 저장부 10, 20, 30: 파라미터 공유 장치

11: 딥러닝 모델 복제부 12: 분산 딥러닝 학습 엔진부

13: CPU 장치 접근부 14: GPU 장치 접근부

100,200, 300: 메모리 박스 접근부 110: 메모리 할당부

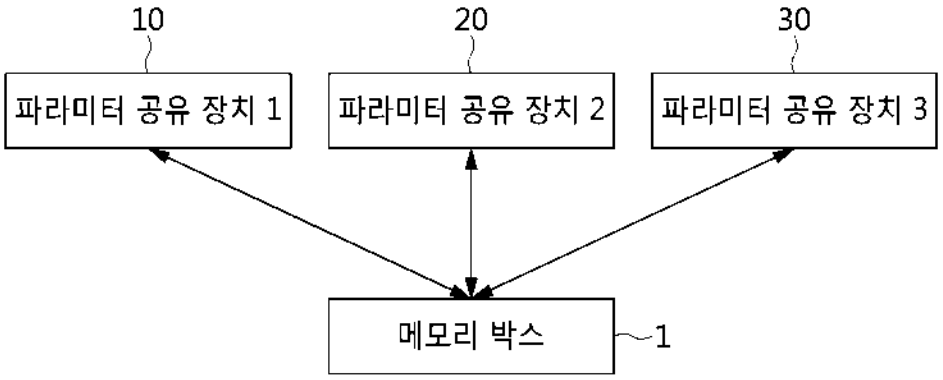
111: 파라미터 정보 수신부 112: 매핑 테이블 관리부

113: 공유 메모리 관리부 120: 연산 처리부

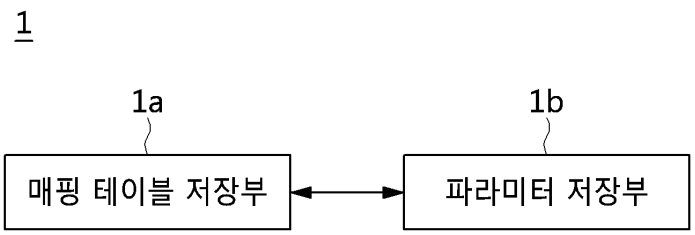
121: 파라미터 객체부 122: 파라미터 처리부

도면

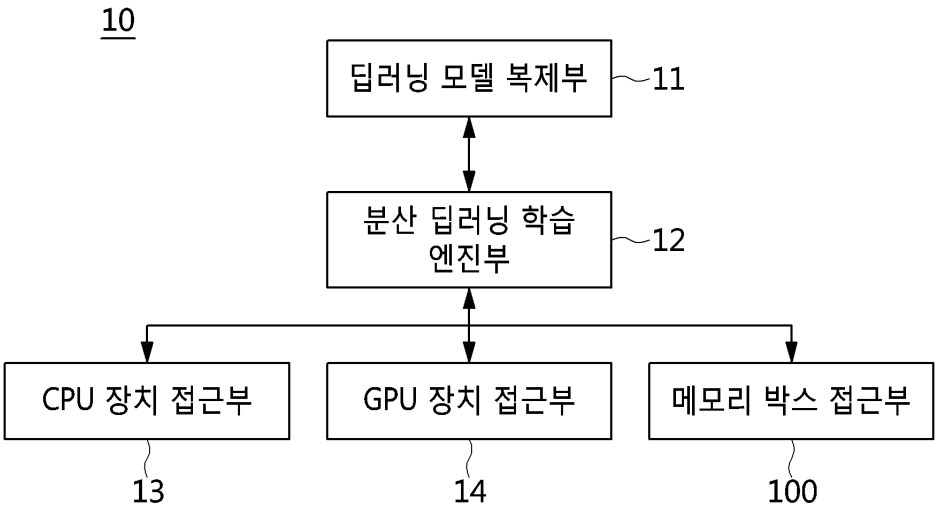
도면1



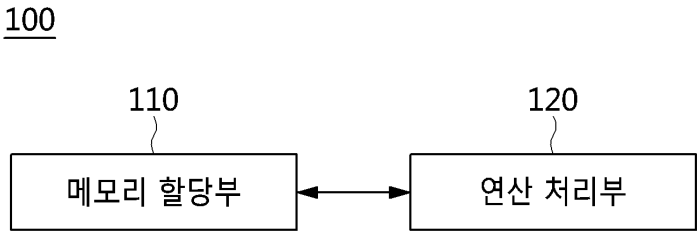
도면2



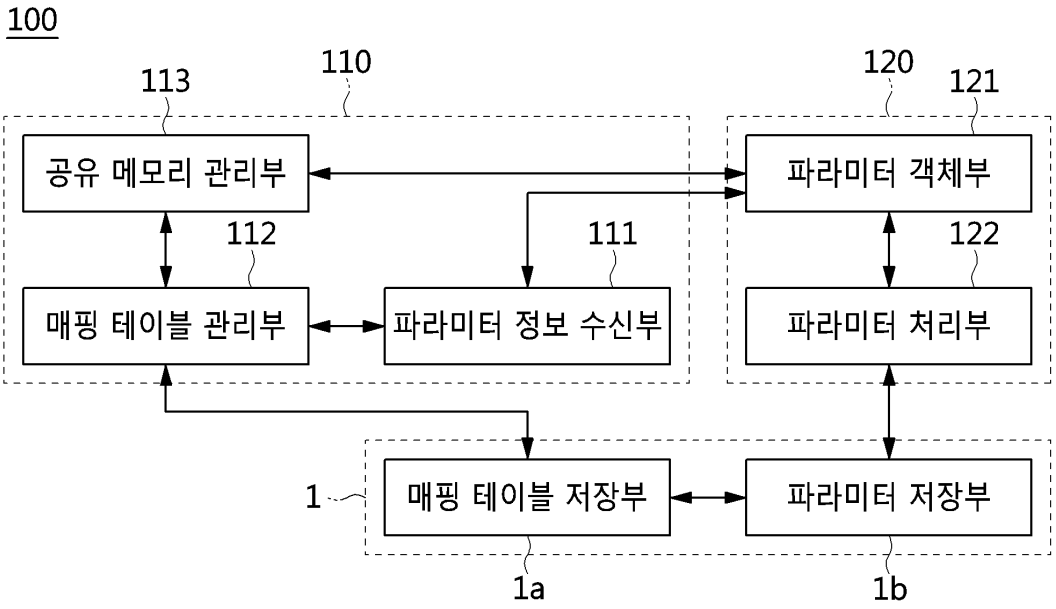
도면3


10
딥러닝 모델 복제부
11
분산 딥러닝 학습 엔진부
12
CPU 장치 접근부
GPU 장치 접근부
메모리 박스 접근부
13
14
100

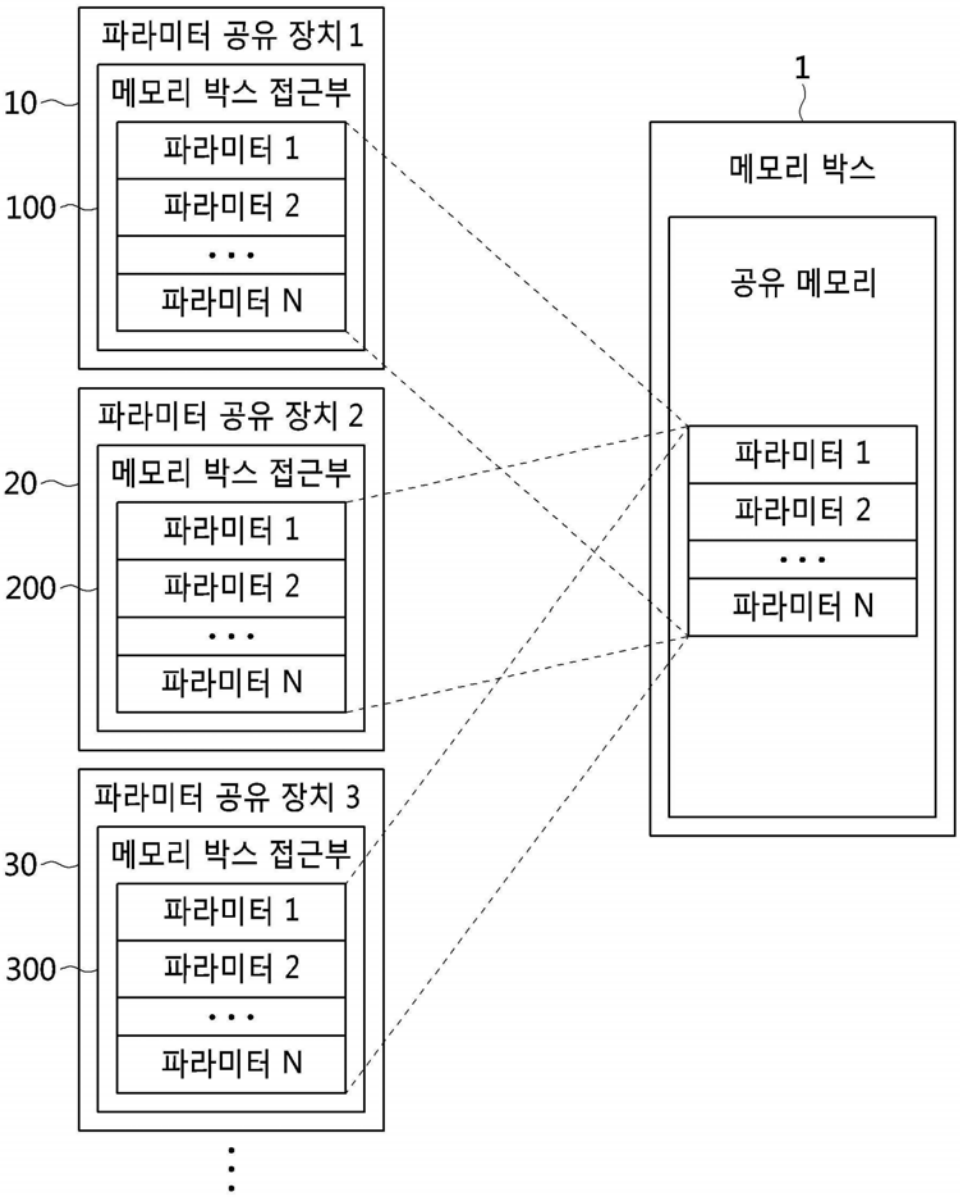
도면4


100
110
120
메모리 할당부
연산 처리부

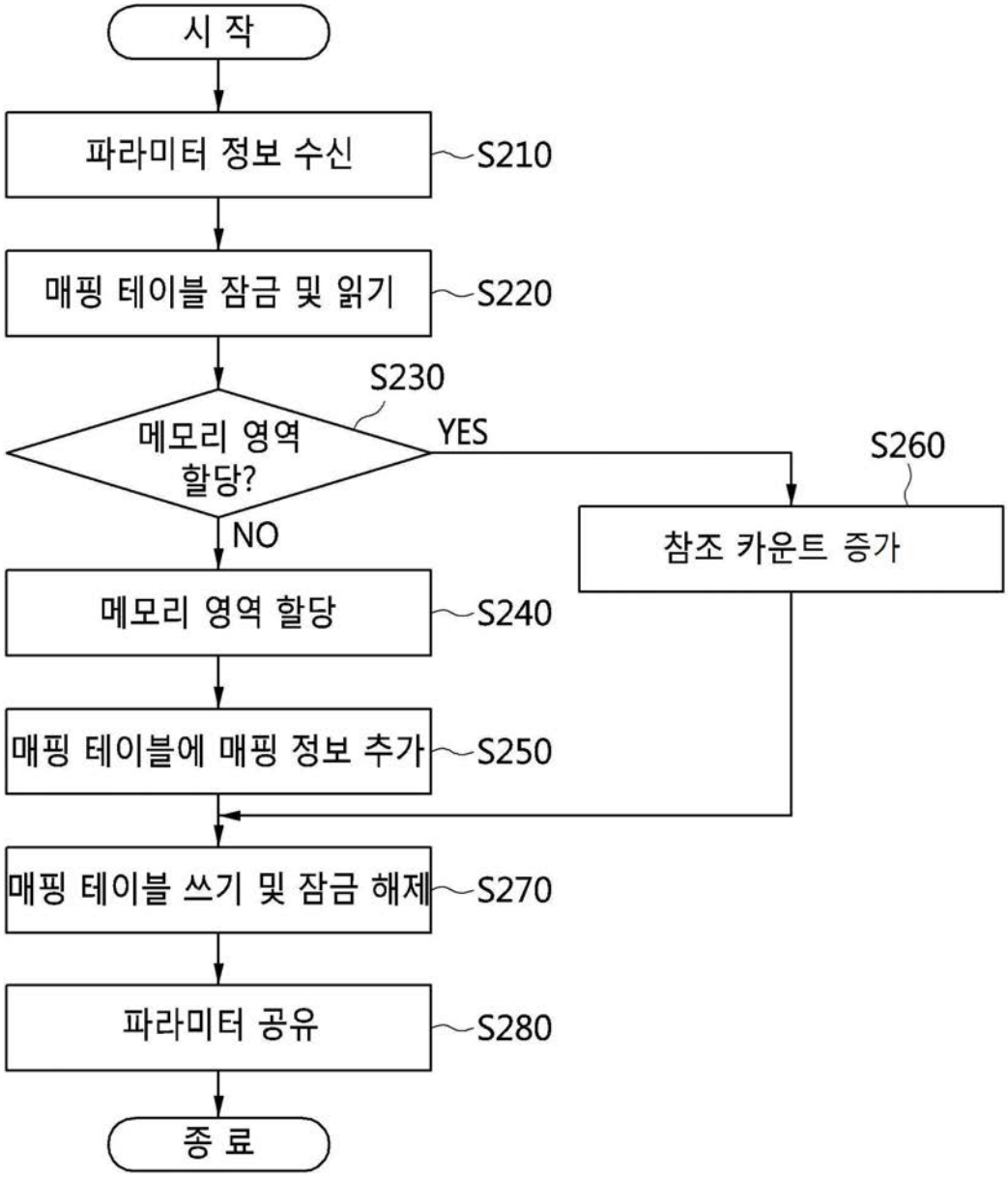
도면5


100
113
110
120
121
공유 메모리 관리부
파라미터 객체부
112
111
122
매핑 테이블 관리부
파라미터 정보 수신부
파라미터 처리부
1
매핑 테이블 저장부
파라미터 저장부
1a
1b

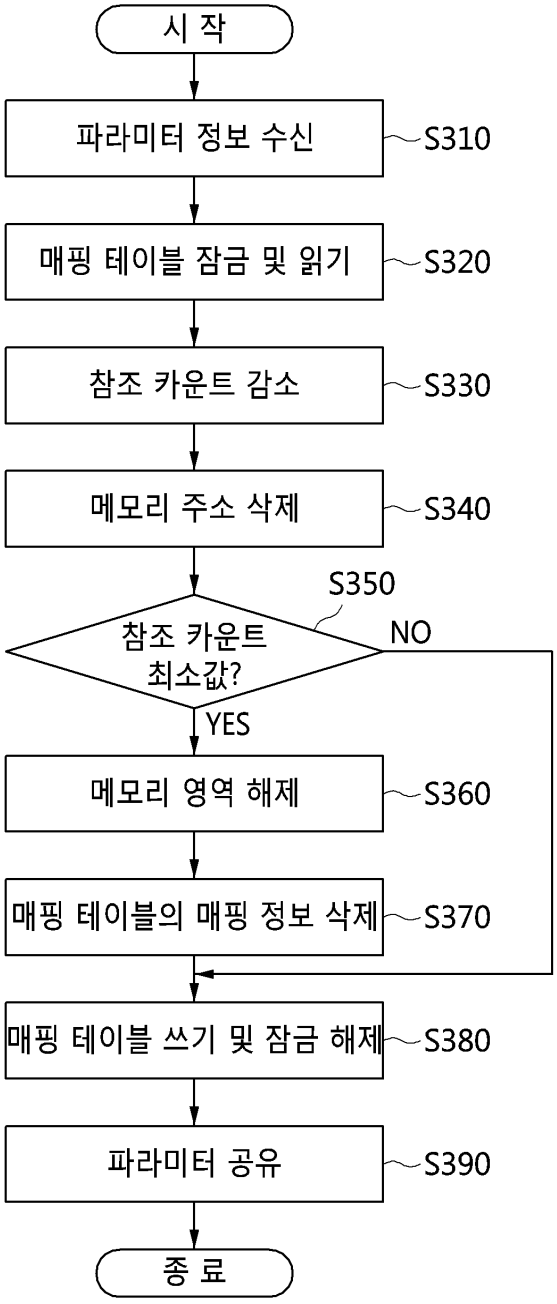
도면6


파라미터 공유 장치 1
10
메모리 박스 접근부
100
파라미터 1
파라미터 2
• • •
파라미터 N
파라미터 공유 장치 2
20
메모리 박스 접근부
200
파라미터 1
파라미터 2
• • •
파라미터 N
파라미터 공유 장치 3
30
메모리 박스 접근부
300
파라미터 1
파라미터 2
• • •
파라미터 N
1
메모리 박스
공유 메모리
파라미터 1
파라미터 2
• • •
파라미터 N

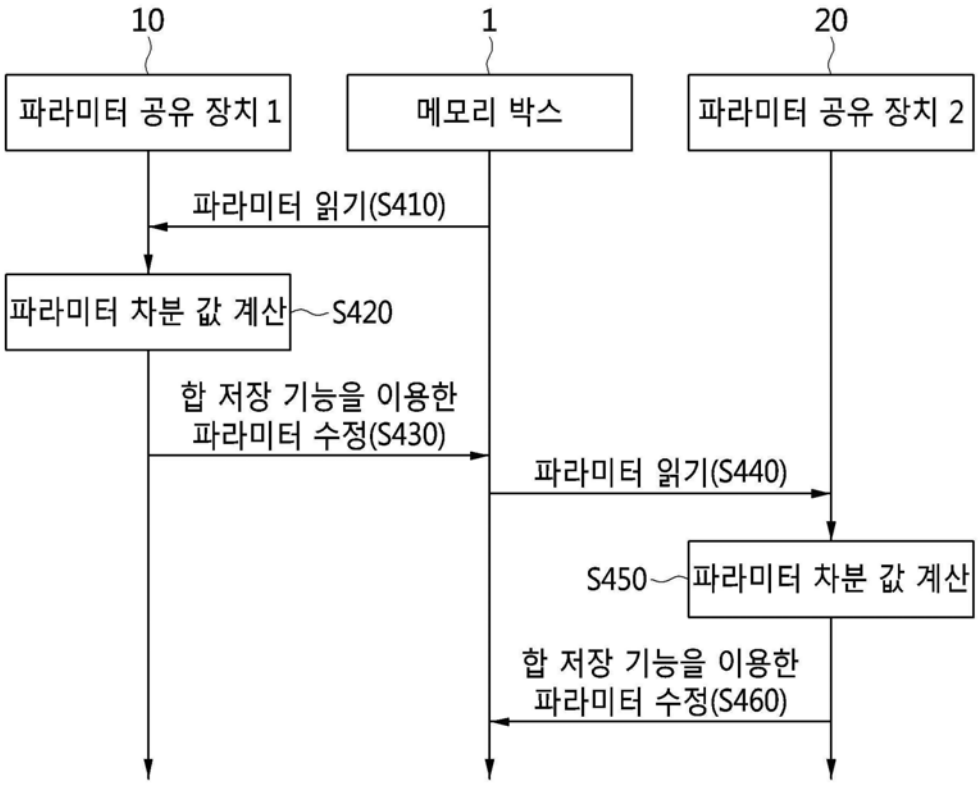
도면7


시 작
파라미터 정보 수신
S210
매핑 테이블 잠금 및 읽기
S220
S230
메모리 영역 할당?
YES
NO
S260
참조 카운트 증가
메모리 영역 할당
S240
매핑 테이블에 매핑 정보 추가
S250
매핑 테이블 쓰기 및 잠금 해제
S270
파라미터 공유
S280
종 료

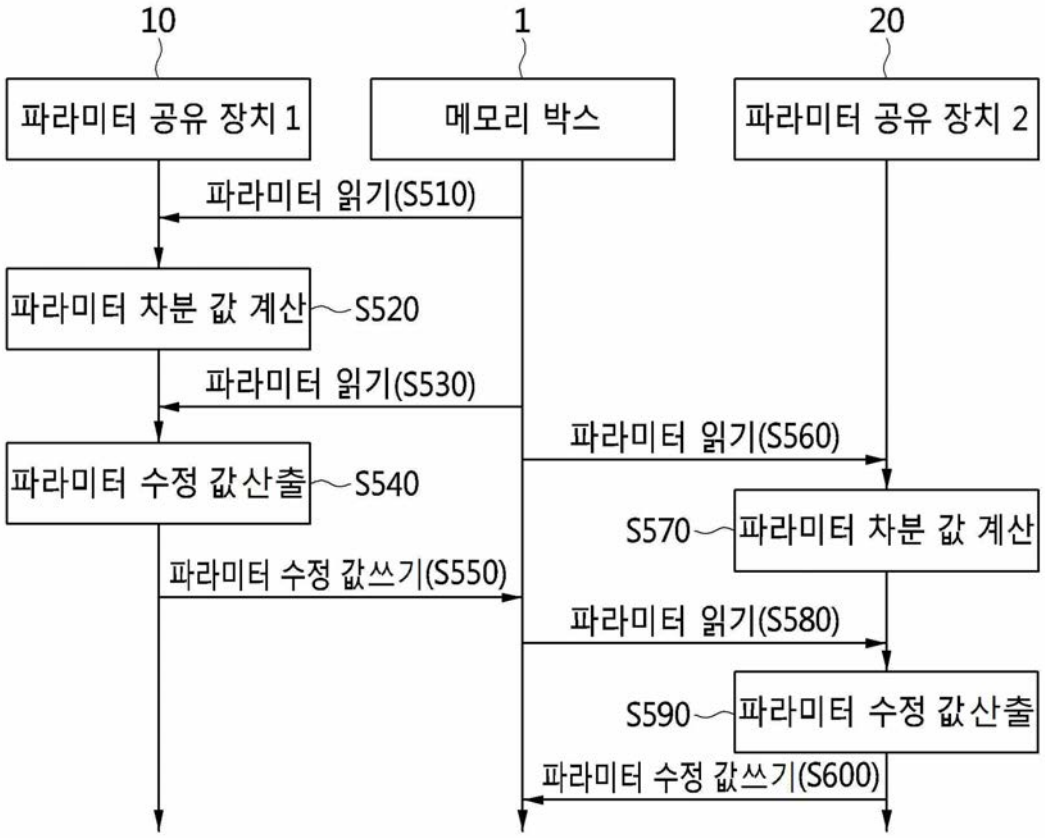
도면8


시 작
파라미터 정보 수신
S310
매핑 테이블 잠금 및 읽기
S320
참조 카운트 감소
S330
메모리 주소 삭제
S340
S350
참조 카운트
최소값?
NO
YES
메모리 영역 해제
S360
매핑 테이블의 매핑 정보 삭제
S370
매핑 테이블 쓰기 및 잠금 해제
S380
파라미터 공유
S390
종 료

도면9


10
1
20
파라미터 공유 장치 1
메모리 박스
파라미터 공유 장치 2
파라미터 읽기(S410)
파라미터 차분 값 계산
S420
합 저장 기능을 이용한
파라미터 수정(S430)
파라미터 읽기(S440)
S450
파라미터 차분 값 계산
합 저장 기능을 이용한
파라미터 수정(S460)

도면10


10
1
20
파라미터 공유 장치 1
메모리 박스
파라미터 공유 장치 2
파라미터 읽기(S510)
파라미터 차분 값 계산
S520
파라미터 읽기(S530)
파라미터 읽기(S560)
파라미터 수정 값산출
S540
S570
파라미터 차분 값 계산
파라미터 수정 값쓰기(S550)
파라미터 읽기(S580)
S590
파라미터 수정 값산출
파라미터 수정 값쓰기(S600)

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년12월31일
(11) 등록번호 10-2197247
(24) 등록일자 2020년12월24일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2006.01) *G06N 3/04* (2006.01)
(52) CPC특허분류
G06N 3/08 (2013.01)
G06N 3/04 (2013.01)
(21) 출원번호 10-2017-0068445
(22) 출원일자 2017년06월01일
심사청구일자 2019년03월14일
(65) 공개번호 10-2018-0131836
(43) 공개일자 2018년12월11일
(56) 선행기술조사문헌
KR1020180051987 A
(뒷면에 계속)
전체 청구항 수 : 총 16 항

(73) 특허권자
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
(72) 발명자
안신영
대전광역시 서구 둔산북로 160, 5동 701호
임은지
대전광역시 유성구 노은동로 187, 602동 1801호
(뒷면에 계속)
(74) 대리인
한양특허법인

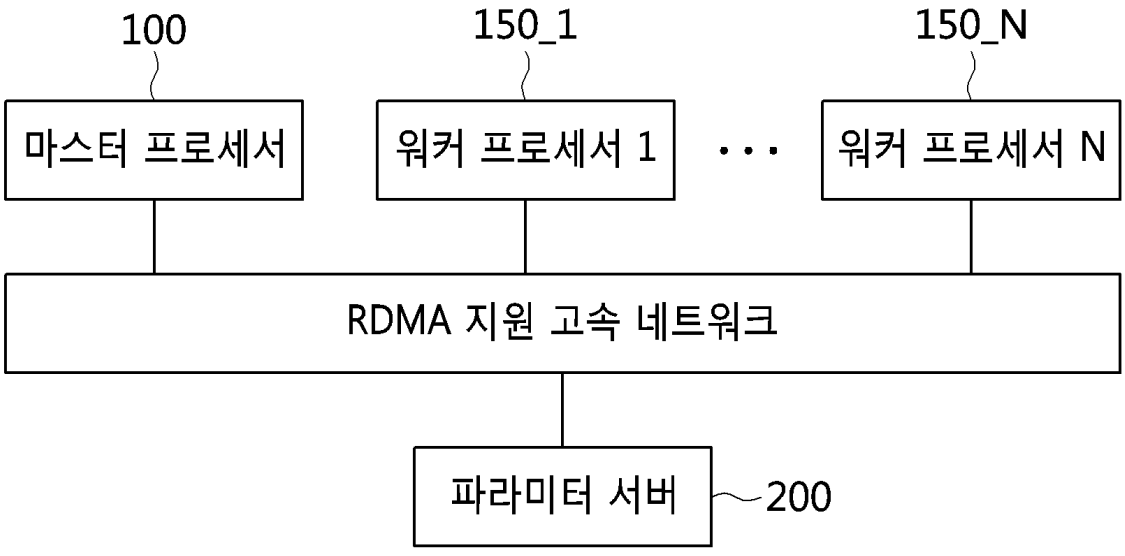
심사관 : 송근배

(54) 발명의 명칭 **파라미터 서버 및 그것에 의해 수행되는 분산 딥러닝 파라미터 공유 방법**

(57) 요 약

파라미터 서버 및 그것에 의해 수행되는 분산 딥러닝 파라미터 공유 방법이 개시된다. 본 발명에 따른 파라미터 서버에 의해 수행되는 분산 딥러닝 파라미터 공유 방법은, 마스터 프로세스의 초기화 요청에 상응하도록 전역 가중치 파라미터를 초기화하는 단계, 로컬 가중치 파라미터를 상기 전역 가중치 파라미터로 업데이트한 후 딥러닝 트레이닝을 수행한 상기 워커 프로세스로부터, 학습된 로컬 그래디언트 파라미터를 입력받아 업데이트하는 단계, 상기 마스터 프로세스의 요청에 따라, 그래디언트 파라미터 누적을 연산하는 단계, 그리고 상기 하나 이상의 워커 프로세스의 상기 그래디언트 파라미터 누적을 이용하여 전역 가중치 파라미터를 계산한 상기 마스터 프로세스로부터, 상기 전역 가중치 파라미터를 입력받아 업데이트하는 단계를 포함한다.

대 표 도 - 도1



(72) 발명자

최용석

전광역시 유성구 지족북로 60, 207동 303호

우영춘

대전광역시 유성구 어은로 57, 113동 404호

최완

대전광역시 서구 관저북로 52, 108동 306호

(56) 선행기술조사문헌

KR1020180125734 A

US20150324690 A1*

US20160103901 A1

US20170098171 A1

US20180218257 A1

Deep Image, Scaling up Image Recognition. Ren Wu et al. Baidu. 2015.02.06.*

*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호	R7117-16-0235
부처명	미래창조과학부
과제관리(전문)기관명	정보통신기술진흥센터(IITP)
연구사업명	정보통신방송기술개발사업(SW컴퓨팅 산업원천기술개발사업)
연구과제명	대규모 딥러닝 고속 처리를 위한 HPC 시스템 개발
기 여 율	1/1
과제수행기관명	한국전자통신연구원
연구기간	2016.04.01 ~ 2016.12.31

명 세 서

청구범위

청구항 1

파라미터 서버에 의해 수행되는 분산 딥러닝 파라미터 공유 방법에 있어서,

마스터 프로세스 및 적어도 하나의 워커 프로세스를 포함하는 분산 딥러닝 프로세스의 요청에 상응하도록, 원격 공유 메모리를 생성 및 할당하는 단계,

상기 원격 공유 메모리의 마스터 가중치 파라미터 영역을 초기화하는 단계,

상기 분산 딥러닝 프로세스들이 상기 원격 공유 메모리를 통해 공유한 분산 딥러닝 파라미터를 이용하여 분산 딥러닝 트레이닝을 수행하는 단계, 그리고

상기 분산 딥러닝 트레이닝의 수행이 완료된 후, 사용이 완료된 상기 원격 공유 메모리를 해제 및 삭제하는 단계를 포함하고,

상기 원격 공유 메모리를 생성 및 할당하는 단계는,

상기 마스터 프로세스의 요청에 따라 원격 공유 메모리를 생성하고, 상기 적어도 하나의 워커 프로세스에게 상기 원격 공유 메모리에 접근하기 위한 접근 정보를 전달하고,

상기 마스터 프로세스와 상기 적어도 하나의 워커 프로세스가, 상기 원격 공유 메모리에 상응하는 각각의 로컬 물리 메모리를 할당하고, 상기 각각의 로컬 물리 메모리를 분산 딥러닝 트레이닝 엔진의 가상 주소 공간에 맵핑하고,

상기 분산 딥러닝 트레이닝을 수행하는 단계는

상기 파라미터 서버, 상기 마스터 프로세스 및 상기 적어도 하나의 워커 프로세스가, 상기 각각의 로컬 물리 메모리와 상기 원격 공유 메모리의 명시적 동기화를 통해 상기 분산 딥러닝 파라미터를 공유하여 상기 분산 딥러닝 트레이닝을 수행하고,

상기 원격 공유 메모리는

상기 마스터 프로세스에 의해 생성되고, 마스터 가중치 파라미터와 마스터 그래디언트 파라미터를 저장하는 마스터 영역; 및

상기 적어도 하나의 워커 프로세스의 개수에 상응하도록 생성되고, 적어도 하나의 워커 그래디언트 파라미터를 각각 저장하는 적어도 하나의 워커 영역;

을 포함하고,

상기 분산 딥러닝 트레이닝을 수행하는 단계는

상기 마스터 프로세스가, 상기 마스터 영역에 상기 마스터 가중치 파라미터와 상기 마스터 그래디언트 파라미터를 업데이트하고, 상기 적어도 하나의 워커 프로세스에게 상기 마스터 영역에 접근하기 위한 접근 정보를 전송하고,

상기 적어도 하나의 워커 프로세스가, 상기 접근 정보를 이용하여 상기 마스터 영역에 접근하여 상기 마스터 가중치 파라미터를 자신의 워커 가중치 파라미터로 업데이트하고,

상기 원격 공유 메모리의 적어도 하나의 워커 영역 중 자신이 생성한 워커 영역에 상기 분산 딥러닝 트레이닝을 수행한 결과로부터 학습된 워커 그래디언트 파라미터를 업데이트하고,

상기 파라미터 서버가, 상기 적어도 하나의 워커 프로세스로부터 상기 적어도 하나의 워커 영역에 상기 적어도 하나의 워커 그래디언트 파라미터가 업데이트되었음을 알림받으면, 상기 적어도 하나의 워커 그래디언트 파라미터를 상기 마스터 그래디언트 파라미터에 업데이트하고,

상기 마스터 프로세스가, 업데이트된 상기 마스터 그래디언트 파라미터를 이용하여 상기 마스터 가중치 파라미

터를 업데이트하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

청구항 2

제1항에 있어서,

상기 원격 공유 메모리를 생성 및 할당하는 단계는,

상기 마스터 프로세스로부터 상기 분산 딥러닝 파라미터를 저장하기 위한 상기 원격 공유 메모리의 생성 요청을 수신하는 단계,

상기 원격 공유 메모리의 생성 요청에 상응하도록 상기 원격 공유 메모리를 생성하는 단계,

생성된 상기 원격 공유 메모리에 상응하는 원격 공유 메모리 생성키 및 상기 원격 공유 메모리에 접근하기 위한 접근 정보를 상기 마스터 프로세스로 전송하는 단계,

상기 마스터 프로세스로부터 상기 원격 공유 메모리에 할당된 원격 공유 메모리 영역에서의 상기 분산 딥러닝 파라미터의 업데이트 및 상기 분산 딥러닝 파라미터의 연산 완료 여부 중 적어도 하나와 관련된 이벤트를 설정하는 이벤트 설정 요청을 수신하여, 상기 원격 공유 메모리에 상기 이벤트를 설정하는 단계,

상기 마스터 프로세스로부터 상기 원격 공유 메모리 생성키를 전달받은 상기 워커 프로세스로부터, 상기 원격 공유 메모리의 할당 요청을 수신하는 단계, 그리고

상기 원격 공유 메모리의 할당 요청에 상응하도록 상기 원격 공유 메모리에 상기 분산 딥러닝 파리미터를 공유하기 위한 원격 공유 메모리 영역을 할당하며, 할당된 상기 원격 공유 메모리의 원격 공유 메모리 영역에 접근하기 위한 접근 정보를 상기 워커 프로세스로 전송하는 단계를 포함하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

청구항 3

제1항에 있어서,

상기 원격 공유 메모리를 해제 및 삭제하는 단계는,

상기 워커 프로세스로부터 원격 공유 메모리 해제 요청을 수신하여, 상기 원격 공유 메모리를 해제하는 단계,

상기 원격 공유 메모리의 해제 완료 시 상기 마스터 프로세스로부터, 원격 공유 메모리 삭제 요청을 수신하는 단계, 그리고

상기 원격 공유 메모리 삭제 요청에 상응하도록 상기 원격 공유 메모리를 삭제하는 단계를 더 포함하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

청구항 4

제1항에 있어서,

상기 분산 딥러닝 트레이닝을 수행하도록 하는 단계는,

상기 분산 딥러닝 프로세스들이 상기 원격 공유 메모리를 통하여 동기식 또는 비동기식으로 업데이트된 딥러닝 파라미터를 공유하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

청구항 5

제4항에 있어서,

상기 동기식으로 업데이트된 상기 딥러닝 파라미터를 공유하여, 상기 분산 딥러닝 트레이닝을 수행하도록 하는 단계는,

상기 분산 딥러닝 프로세스들의 워커 로컬 가중치 파라미터 영역을 상기 원격 공유 메모리의 상기 마스터 가중치 파라미터의 값으로 업데이트하는 단계,

동기 방식으로 분산 딥러닝 트레이닝을 수행한 상기 워커 프로세스로부터, 학습된 워커 로컬 그래디언트 파라미터를 입력받아 그래디언트 파라미터 누적을 연산하는 단계,

상기 하나 이상의 워커 프로세스의 상기 그래디언트 파라미터 누적을 이용하여 마스터 가중치 파라미터를 계산한 상기 마스터 프로세스로부터, 상기 마스터 가중치 파라미터를 입력받아 상기 마스터 가중치 파라미터 영역을 업데이트하는 단계, 그리고

상기 마스터 가중치 파라미터 영역의 업데이트를 적어도 어느 하나의 상기 워커 프로세스로 알리는 단계를 포함하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

청구항 6

제5항에 있어서,

상기 그래디언트 파라미터 누적을 연산하는 단계는,

상기 분산 딥러닝 트레이닝을 수행한 상기 워커 프로세스들이 학습한 워커 로컬 그래디언트 파라미터를 상기 원격 공유 메모리의 워커 그래디언트 파라미터 영역에 저장하는 단계,

상기 워커 프로세스들로부터 그래디언트 파라미터 누적 연산을 요청받는 단계,

요청에 상응하는 상기 원격 공유 메모리의 상기 워커 그래디언트 파라미터를 마스터 그래디언트 파라미터에 누적 연산하는 단계, 그리고

상기 누적 연산의 완료를 상기 마스터 프로세스로 알리는 단계

를 포함하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

청구항 7

제4항에 있어서,

상기 비동기식으로 업데이트된 상기 딥러닝 파라미터를 공유하여, 상기 분산 딥러닝 트레이닝을 수행하도록 하는 단계는,

하나 이상의 상기 워커 프로세스의 워커 로컬 가중치 파라미터 영역을 상기 원격 공유 메모리의 상기 마스터 가중치 파라미터의 값으로 업데이트 하는 단계,

상기 분산 딥러닝 트레이닝을 수행한 상기 하나 이상의 워커 프로세스들이 원격 공유 메모리 상의 워커 그래디언트 파라미터를 업데이트하는 단계,

상기 하나 이상의 워커 프로세스로부터 수신한 마스터 가중치 파라미터의 업데이트 요청에 상응하도록, 상기 마스터 가중치 파라미터 영역을 업데이트하는 단계, 그리고

상기 분산 딥러닝 트레이닝의 수행이 완료된 후, 사용이 완료된 상기 원격 공유 메모리를 해제 및 삭제하는 단계를 포함하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

청구항 8

마스터 프로세스 및 적어도 하나의 워커 프로세스를 포함하는 분산 딥러닝 프로세스의 요청과 관련된 메시지를 송수신하는 통신 처리부,

상기 분산 딥러닝 프로세스의 요청에 상응하도록, 분산 딥러닝 파라미터를 저장하기 위한 원격 공유 메모리를 생성, 할당 및 해제하는 원격 공유 메모리 관리부, 그리고

상기 분산 딥러닝 프로세스가 상기 원격 공유 메모리를 통해 공유한 분산 딥러닝 파라미터를 이용하여 분산 딥러닝 트레이닝을 수행하는 파라미터 연산부를 포함하고,

상기 원격 공유 메모리 관리부는

상기 마스터 프로세스의 요청에 따라 상기 원격 공유 메모리를 생성하고, 상기 적어도 하나의 워커 프로세스에게 상기 원격 공유 메모리에 접근하기 위한 접근 정보를 전달하고,

상기 마스터 프로세스와 상기 적어도 하나의 워커 프로세스는

상기 원격 공유 메모리에 상응하는 각각의 로컬 물리 메모리를 할당하고, 상기 각각의 로컬 물리 메모리를 분산

딥러닝 트레이닝 엔진의 가상 주소 공간에 맵핑하고,

상기 파라미터 연산부는

상기 마스터 프로세스 및 상기 적어도 하나의 워커 프로세스와 함께, 상기 각각의 로컬 물리 메모리와 상기 원격 공유 메모리의 명시적 동기화를 통해 상기 분산 딥러닝 파라미터를 공유하여 상기 분산 딥러닝 트레이닝을 수행하고,

상기 원격 공유 메모리는

상기 마스터 프로세스에 의해 생성되고, 마스터 가중치 파라미터와 마스터 그래디언트 파라미터를 저장하는 마스터 영역; 및

상기 적어도 하나의 워커 프로세스의 개수에 상응하도록 생성되고, 적어도 하나의 워커 그래디언트 파라미터를 각각 저장하는 적어도 하나의 워커 영역;

을 포함하고,

상기 마스터 프로세스는

상기 마스터 영역에 상기 마스터 가중치 파라미터와 상기 마스터 그래디언트 파라미터를 업데이트하고,

상기 적어도 하나의 워커 프로세스에게 상기 마스터 영역에 접근하기 위한 접근 정보를 전송하고,

상기 적어도 하나의 워커 프로세스는

상기 접근 정보를 이용하여 상기 마스터 영역에 접근하여 상기 마스터 가중치 파라미터를 자신의 워커 가중치 파라미터로 업데이트하고,

상기 원격 공유 메모리의 적어도 하나의 워커 영역 중 자신이 생성한 워커 영역에 상기 분산 딥러닝 트레이닝을 수행한 결과로부터 학습된 워커 그래디언트 파라미터를 업데이트하고,

상기 파라미터 연산부는

상기 적어도 하나의 워커 프로세스로부터 상기 적어도 하나의 워커 영역에 상기 적어도 하나의 워커 그래디언트 파라미터가 업데이트되었음을 알림받으면, 상기 적어도 하나의 워커 그래디언트 파라미터를 상기 마스터 그래디언트 파라미터에 누적 연산하여 업데이트하고,

상기 마스터 프로세스는

업데이트된 상기 마스터 그래디언트 파라미터를 이용하여 상기 마스터 가중치 파라미터를 업데이트하는 것을 특징으로 하는 파라미터 서버.

청구항 9

제8항에 있어서,

상기 파라미터 연산부는,

두 개의 원격 공유 메모리 영역에 대한 벡터 연산을 수행하는 것을 특징으로 하는 파라미터 서버.

청구항 10

제9항에 있어서,

상기 파라미터 연산부는,

제1 벡터에 제1 상수를 곱하는 연산, 상기 제1 상수를 곱한 제1 벡터와 제2 벡터를 합하는 연산 및 상기 제1 상수를 곱한 상기 제1 벡터와 제2 상수를 곱한 상기 제2 벡터를 합하는 연산 중 적어도 어느 하나의 상기 벡터 연산을 수행하는 것을 특징으로 하는 파라미터 서버.

청구항 11

제8항에 있어서,

상기 파라미터 연산부는,

가중치 파라미터 및 그래디언트 파라미터 중 적어도 어느 하나를 포함하는 상기 분산 딥러닝 파라미터를 연산하는 것을 특징으로 하는 파라미터 서버.

청구항 12

제11항에 있어서,

상기 마스터 프로세스는,

상기 마스터 프로세스가 할당한 모든 상기 원격 공유 메모리의 영역에 접근 가능하고,

상기 워커 프로세스는,

마스터 파라미터 영역 및 상기 워커 프로세스가 딥러닝 트레이닝을 수행한 결과를 저장하는 워커 파라미터 영역만 접근 가능한 것을 특징으로 하는 파라미터 서버.

청구항 13

제12항에 있어서,

상기 파라미터 연산부는,

동기식으로 상기 분산 딥러닝 파라미터를 공유하는 경우, 상기 그래디언트 파라미터 누적을 연산하는 것을 특징으로 하는 파라미터 서버.

청구항 14

제12항에 있어서,

상기 파라미터 연산부는,

비동기식으로 상기 분산 딥러닝 파라미터를 공유하는 경우, 상기 워커 프로세스로부터 수신한 워커 그래디언트 파라미터를 마스터 가중치 파라미터 영역에 업데이트하는 것을 특징으로 하는 파라미터 서버.

청구항 15

삭제

청구항 16

삭제

청구항 17

제8항에 있어서,

상기 원격 공유 메모리 관리부는,

상기 워커 프로세스로부터 수신한 원격 공유 메모리 해제 요청에 상응하도록 상기 원격 공유 메모리를 해제하고, 상기 마스터 프로세스로부터 수신한 원격 공유 메모리 삭제 요청에 상응하도록 상기 원격 공유 메모리를 삭제하는 것을 특징으로 하는 파라미터 서버.

청구항 18

제8항에 있어서,

상기 마스터 프로세스 및 워커 프로세스는,

원격 직접 메모리 접근(RDMA)을 지원하는 고속 네트워크를 통하여, 상기 파라미터 서버에 저장한 상기 분산 딥러닝 파라미터를 직접 읽어오거나 쓰는 방식으로 상기 분산 딥러닝 파라미터를 공유하는 것을 특징으로 하는 파라미터 서버.

발명의 설명

기 술 분 야

[0001] 본 발명은 분산 딥러닝 프레임워크에서 분산 트레이닝되는 파라미터를 공유하는 기술에 관한 것으로, 특히 분산 딥러닝 프로세스들이 파라미터 서버의 물리 메모리를 공유 메모리 형태로 접근할 수 있도록 하여, 딥러닝 프로세스들 간 파라미터 공유를 가속화하는 기술에 관한 것이다.

배 경 기 술

[0002] 딥러닝이란 사람의 신경세포(Biological Neuron)를 모사하여 기계가 학습하도록 하는 인공신경망(Artificial Neural Network) 기반의 기계 학습법을 의미한다. 최근, 딥러닝 기술은 이미지 인식, 음성 인식, 자연어 처리의 발전에 기여하며 주목 받고 있다. 그리고 오늘날의 딥러닝 모델들은 응용의 인식 성능을 높이기 위해 모델의 계층이 깊어지고(Deep), 특징(Feature)이 많아지는(Wide) 대규모 모델로 진화하고 있다.

[0003] 그러나 대형화되는 딥러닝 모델과 대규모의 학습 데이터를 단일 머신에서 처리하기에는 한계가 있다. 이에, 대규모 분산 컴퓨팅 자원을 활용하려는 노력의 일환으로 딥러닝 분산 플랫폼 기술이 개발되고 있다.

[0004] 딥러닝 분산 플랫폼에서는 분산 병렬 처리를 통하여 딥러닝 트레이닝 가속을 시도하는데, 분산 병렬 처리 방법으로 데이터 병렬 처리(Data Parallelism)와 모델 병렬 처리(Model Parallelism) 방법이 있다. 데이터 병렬 처리란 학습해야 하는 입력 데이터 집합을 다수의 컴퓨터들이 나누어 트레이닝하는 방법이고, 모델 병렬 처리란 딥러닝 모델을 나누어 다수의 컴퓨터들이 트레이닝하는 방법이다.

[0005] 딥러닝 트레이닝 분산 병렬 처리 시에는 트레이닝의 대상이 되는 가중치와 특징값 등의 파라미터들을 모든 컴퓨터가 공유해야 한다. 파라미터를 공유하는 방법에는 각 컴퓨터들이 다른 모든 컴퓨터들에게 직접 파라미터를 전달하는 풀 메시(full mesh) 토폴로지 기반 공유 방법과 모든 분산 컴퓨터들이 공유 장소를 이용하여 파라미터를 읽고 쓰는 스타(star) 토폴로지 기반의 공유 방법이 있다. 그리고 대부분의 분산 플랫폼은 일반적으로 중앙 집중형 파라미터 공유 저장소(파라미터 서버)를 통해 파라미터를 교환하는 두 번째 방식을 선택하고 있다.

[0006] 파라미터 공유 방법에서는 분산된 컴퓨터들이 각각 파라미터를 중앙 집중형으로 업데이트 하기 때문에 가중치를 갱신해야 하는 주기(일정 트레이닝 반복)마다 분산 트레이닝 중인 컴퓨터 간 파라미터 동기화가 필요하다. 동기식 업데이트의 경우는 딥러닝을 분산 처리하는 컴퓨터들의 일정 트레이닝 반복마다 파라미터를 파라미터 서버로 전송하여 분산 트레이닝된 파라미터를 통합한다.

[0007] 반면, 비동기식 업데이트 방식은 파라미터 서버가 분산 컴퓨터들로부터 늦거나 빨리 도착하는 파라미터들의 동기를 맞추지 않고 트레이닝을 진행하는 방법이다. 비동기 방식은 동기식에 비해 정확성을 크게 희생시키지 않으면서 빠르게 트레이닝 할 수 있는 장점이 있다. 대부분의 분산 프레임워크들에서는 동기식 방법과 비동기식 방법을 모두 또는 선택적으로 제공하고 있다.

[0008] 각 딥러닝 분산 플랫폼에서 파라미터 서버를 구현하는 방법으로는 마스터 역할을 하는 프로세스가 자신의 메모리에 마스터 파라미터를 저장하는 영역을 할당한다. 그리고 분산 트레이닝을 수행하는 워커(또는 슬레이브) 프로세스들이 통신 메시지 형태로 주기적으로 보내주는 파라미터로 마스터 파라미터를 업데이트 한 후 다시 워커 프로세스들에 업데이트된 마스터 파라미터를 배포하는 방식이 있다. Petuum, CNTK와 같은 분산 플랫폼은 파라미터 서버 전용 목적으로 개발된 분산 키-밸류 저장소를 이용하기도 한다.

[0009] 종래 기술에 따르면, 파라미터 서버와 분산 컴퓨터간에 메시지 송수신 형태의 통신방법을 통해 파라미터를 교환한다. 그러나, 메시지 송수신 형태의 통신 방법으로 파라미터를 교환할 경우, 통신 오버헤드가 높고, CPU, GPU 등이 대기하는 시간이 길어지며, 이는 자원 사용률의 저하로 이어진다.

[0010] 따라서, 통신 프로토콜로 대규모 파라미터를 송수신하는 기술의 한계를 극복하여, 추가적인 메모리 복사 및 프로토콜 처리 등의 통신 오버헤드를 대폭 경감하고, 통신 성능을 개선할 수 있는 파라미터 공유 기술의 개발이 필요하다.

선행기술문헌

특허문헌

[0011] (특허문헌 0001) 한국 등록 특허 제10-1559089호, 2015년 10월 02일 공개(명칭: 장치의 컴포넌트들 간에 메모리

자원들을 공유하기 위한 통신 프로토콜)

발명의 내용

해결하려는 과제

[0012] 본 발명의 목적은 분산 딥러닝 플랫폼에서 분산 트레이닝을 수행하는 프로세스들이 대규모 파라미터를 교환할 수 있도록 하는 것이다.

[0013] 또한, 본 발명의 목적은 파라미터 서버와 분산 컴퓨터가 메시지 송수신 형태의 통신 방법으로 파라미터를 교환할 경우 발생하는 추가적인 메모리 복사 및 통신 오버헤드를 대폭 경감하는 것이다.

[0014] 또한, 본 발명의 목적은 메시지 송수신 형태의 통신 방법으로 파라미터를 교환하는 방법에 비하여, 통신 성능을 개선하고, 파라미터 송수신 시 유휴 상태인 계산 자원의 이용률을 극대화하는 것이다.

과제의 해결 수단

[0015] 상기한 목적을 달성하기 위한 본 발명에 따른 파라미터 서버에 의해 수행되는 분산 딥러닝 파라미터 공유 방법은 마스터 프로세스 및 워커 프로세스 중 적어도 어느 하나를 포함하는 분산 딥러닝 프로세스의 요청에 상응하도록, 공유 메모리를 생성 및 할당하는 단계, 상기 공유 메모리의 마스터 가중치 파라미터 영역을 초기화하는 단계, 상기 분산 딥러닝 프로세스들이 상기 공유 메모리를 통해 공유한 딥러닝 파라미터를 이용하여, 분산 딥러닝 트레이닝을 수행하도록 하는 단계, 그리고 상기 분산 딥러닝 트레이닝의 수행이 완료된 후, 사용이 완료된 상기 공유 메모리를 해제 및 삭제하는 단계를 포함한다.

[0016] 이때, 상기 공유 메모리를 생성 및 할당하는 단계는, 상기 마스터 프로세스로부터 파라미터용 원격 공유 메모리 생성 요청을 수신하는 단계, 상기 파라미터용 원격 공유 메모리 생성 요청에 상응하도록 공유 메모리를 생성하는 단계, 생성된 상기 공유 메모리에 상응하는 공유 메모리 생성키 및 접근 정보를 상기 마스터 프로세스로 전송하는 단계, 상기 마스터 프로세스로부터 이벤트 설정 요청을 수신하여, 상기 공유 메모리의 이벤트를 설정하는 단계, 상기 마스터 프로세스로부터 상기 공유 메모리 생성키를 전달받은 상기 워커 프로세스로부터, 공유 메모리 할당 요청을 수신하는 단계, 그리고 상기 공유 메모리를 할당하고, 할당된 상기 공유 메모리의 접근 정보를 상기 워커 프로세스로 전송하는 단계를 더 포함할 수 있다.

[0017] 이때, 상기 공유 메모리를 해제 및 삭제하는 단계는, 상기 워커 프로세스로부터 공유 메모리 해제 요청을 수신하여, 상기 공유 메모리를 해제하는 단계, 상기 공유 메모리의 해제 완료 시 상기 마스터 프로세스로부터, 공유 메모리 삭제 요청을 수신하는 단계, 그리고 상기 공유 메모리 삭제 요청에 상응하도록 상기 공유 메모리를 삭제하는 단계를 더 포함할 수 있다.

[0018] 이때, 상기 분산 딥러닝 트레이닝을 수행하도록 하는 단계는, 상기 분산 딥러닝 프로세스들이 상기 공유 메모리를 통하여 동기식 또는 비동기식으로 업데이트된 가중치 파라미터를 공유할 수 있다.

[0019] 이때, 상기 분산 딥러닝 프로세스들이 상기 공유 메모리를 통해 공유한 딥러닝 파라미터를 이용하여, 동기식 분산 딥러닝 트레이닝을 수행하도록 하는 단계는, 상기 분산 딥러닝 프로세스들의 워커 로컬 가중치 파라미터 영역을 상기 공유 메모리의 상기 마스터 가중치 파라미터의 값으로 업데이트하는 단계, 동기 방식으로 분산 딥러닝 트레이닝을 수행한 상기 워커 프로세스로부터, 학습된 워커 로컬 그래디언트 파라미터를 입력받아 그래디언트 파라미터 누적을 연산하는 단계, 상기 하나 이상의 워커 프로세스의 상기 그래디언트 파라미터 누적을 이용하여 마스터 가중치 파라미터를 계산한 상기 마스터 프로세스로부터, 상기 마스터 가중치 파라미터를 입력받아 상기 마스터 가중치 파라미터 영역을 업데이트하는 단계, 그리고 상기 마스터 가중치 파라미터 영역의 업데이트를 적어도 어느 하나의 상기 워커 프로세스로 알리는 단계를 포함할 수 있다.

[0020] 이때, 상기 그래디언트 파라미터 누적을 연산하는 단계는, 상기 분산 딥러닝 트레이닝을 수행한 상기 워커 프로세스들이 학습한 워커 로컬 그래디언트 파라미터를 상기 공유 메모리의 워커 그래디언트 파라미터 영역에 저장하는 단계, 상기 워커 프로세스들로부터 그래디언트 파라미터 누적 연산을 요청받는 단계, 요청에 상응하는 상기 공유 메모리의 상기 워커 그래디언트 파라미터를 마스터 그래디언트 파라미터에 누적 연산하는 단계, 그리고 상기 누적 연산의 완료를 상기 마스터 프로세스로 알리는 단계를 포함한다.

[0021] 이때, 상기 분산 딥러닝 프로세스들이 상기 공유 메모리를 통해 공유한 딥러닝 파라미터를 이용하여, 비동기식

분산 딥러닝 트레이닝을 수행하도록 하는 단계는, 하나 이상의 상기 워커 프로세스의 워커 로컬 가중치 파라미터 영역을 상기 공유 메모리의 상기 마스터 가중치 파라미터의 값으로 업데이트 하는 단계, 상기 분산 딥러닝 트레이닝을 수행한 상기 하나 이상의 워커 프로세스들이 공유 메모리 상의 워커 그래디언트 파라미터를 업데이트하는 단계, 상기 하나 이상의 워커 프로세스로부터 수신한 마스터 가중치 파라미터의 업데이트 요청에 상응하도록, 상기 마스터 가중치 파라미터 영역을 업데이트하는 단계, 그리고 상기 분산 딥러닝 트레이닝의 수행이 완료된 후, 사용이 완료된 상기 공유 메모리를 해제 및 삭제하는 단계를 포함한다.

[0022] 또한, 본 발명의 일실시예에 따른 파라미터 서버는 마스터 프로세스 및 워커 프로세스 중 적어도 어느 하나와 메시지를 송수신하고, 원격 직접 메모리 접근(RDMA) 방식의 읽기 및 쓰기를 지원하는 통신 처리부, 공유 메모리의 할당 및 해제를 관리하는 공유 메모리 관리부, 분산 딥러닝 파라미터를 계산하는 파라미터 연산부, 그리고 상기 공유 메모리에 대한 이벤트 발생 시, 상기 공유 메모리에 상응하는 상기 마스터 프로세스 및 하나 이상의 상기 워커 프로세스 중 적어도 어느 하나로 상기 이벤트의 발생을 알리는 이벤트 처리부를 포함한다.

[0023] 이때, 상기 파라미터 연산부는, 두 개의 공유 메모리 영역에 대한 벡터/매트릭스 연산을 수행할 수 있다.

[0024] 이때, 상기 파라미터 연산부는, 제1 벡터에 제1 상수를 곱하는 연산, 상기 제1 상수를 곱한 제1 벡터와 제2 벡터를 합하는 연산 및 상기 제1 상수를 곱한 상기 제1 벡터와 제2 상수를 곱한 상기 제2 벡터를 합하는 연산 중 적어도 어느 하나의 상기 벡터 연산을 수행할 수 있다.

[0025] 이때, 상기 파라미터 연산부는, 가중치 파라미터 및 그래디언트 파라미터 중 적어도 어느 하나를 포함하는 상기 분산 딥러닝 파라미터를 연산할 수 있다.

[0026] 이때, 상기 마스터 프로세스는, 상기 마스터 프로세스가 할당한 모든 상기 공유 메모리의 영역에 접근 가능하고, 상기 워커 프로세스는, 마스터 파라미터 영역 및 상기 워커 프로세스가 딥러닝 트레이닝을 수행한 결과를 저장하는 워커 파라미터 영역만 접근 가능할 수 있다.

[0027] 이때, 상기 파라미터 연산부는, 동기식으로 상기 분산 딥러닝 파라미터를 공유하는 경우, 상기 그래디언트 파라미터 누적을 연산할 수 있다.

[0028] 이때, 상기 파라미터 연산부는, 비동기식으로 상기 분산 딥러닝 파라미터를 공유하는 경우, 상기 워커 프로세스로부터 수신한 워커 그래디언트 파라미터를 마스터 가중치 파라미터 영역에 업데이트할 수 있다.

[0029] 이때, 상기 공유 메모리 관리부는, 상기 마스터 프로세스로부터 수신한 파라미터용 원격 공유 메모리 생성 요청에 상응하도록 공유 메모리를 생성하고, 상기 공유 메모리의 공유 메모리 생성키 및 접근 정보를 상기 마스터 프로세스로 전송할 수 있다.

[0030] 이때, 상기 공유 메모리 관리부는, 상기 마스터 프로세스로부터 상기 공유 메모리 생성키를 전달받은 상기 워커 프로세스로부터 공유 메모리 할당 요청을 수신하고, 상기 공유 메모리 할당 요청에 상응하도록 상기 공유 메모리를 할당하며, 할당된 상기 공유 메모리의 접근 정보를 상기 워커 프로세스로 전송할 수 있다.

[0031] 이때, 상기 공유 메모리 관리부는, 상기 워커 프로세스로부터 수신한 공유 메모리 해제 요청에 상응하도록 상기 공유 메모리를 해제하고, 상기 마스터 프로세스로부터 수신한 공유 메모리 삭제 요청에 상응하도록 상기 공유 메모리를 삭제할 수 있다.

[0032] 이때, 상기 마스터 프로세스 및 워커 프로세스는, 상기 원격 직접 메모리 접근(RDMA)을 지원하는 고속 네트워크를 통하여, 상기 파라미터 서버에 저장한 상기 분산 딥러닝 파라미터를 직접 읽어오거나 쓰는 방식으로 상기 분산 딥러닝 파라미터를 공유할 수 있다.

발명의 효과

[0033] 본 발명에 따르면, 분산 딥러닝 플랫폼에서 분산 트레이닝을 수행하는 프로세스들이 대규모 파라미터를 교환할 수 있다.

[0034] 또한 본 발명에 따르면, 파라미터 서버와 분산 컴퓨터가 메시지 송수신 형태의 통신 방법으로 파라미터를 교환할 경우 발생하는 추가적인 메모리 복사 및 통신 오버헤드를 대폭 경감할 수 있다.

[0035] 또한 본 발명에 따르면, 메시지 송수신 형태의 통신 방법으로 파라미터를 교환하는 방법에 비하여, 통신 성능을 개선하고, 파라미터 송수신 시 유휴 상태인 계산 자원의 이용률을 극대화할 수 있다.

도면의 간단한 설명

도 1은 본 발명의 일실시예에 따른 파라미터 서버가 적용되는 분산 딥러닝 프레임워크 환경을 개략적으로 나타낸 도면이다.

도 2는 본 발명의 일실시예에 따른 파라미터 서버의 구성을 나타낸 블록도이다.

도 3은 본 발명의 일실시예에 따른 파라미터 공유를 위한 원격 공유 메모리 가상 매핑 매커니즘을 나타낸 예시도이다.

도 4는 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크의 기능을 설명하기 위한 구조도이다.

도 5는 본 발명의 일실시예에 따른 프로세스별 원격 공유 메모리 할당의 일 예를 나타낸 예시도이다.

도 6은 본 발명의 일실시예에 따른 분산 딥러닝 파라미터 공유 방법을 나타낸 순서도이다.

도 7은 본 발명의 일실시예에 따른 원격 공유 메모리의 생성 및 할당 과정을 나타낸 순서도이다.

도 8은 본 발명의 일실시예에 따른 원격 공유 메모리의 삭제 및 해제 과정을 나타낸 순서도이다.

도 9는 본 발명의 일실시예에 따른 동기식 파라미터 공유 방법을 설명하기 위한 순서도이다.

도 10은 본 발명의 일실시예에 따른 비동기식 파라미터 공유 방법을 설명하기 위한 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0037] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시 예들을 도면에 예시하고 상세하게 설명하고자 한다.

[0038] 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.

[0039] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0040] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가진 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0041] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 본 발명을 설명함에 있어 전체적인 이해를 용이하게 하기 위하여 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.

[0043] 도 1은 본 발명의 일실시예에 따른 파라미터 서버가 적용되는 분산 딥러닝 프레임워크 환경을 개략적으로 나타낸 도면이다.

[0044] 도 1에 도시한 바와 같이, 딥러닝 트레이닝을 수행하는 분산된 계산 노드에서 실행되는 분산 딥러닝 프로세스는 마스터 프로세스(100) 및 하나 이상의 워커 프로세스(150)를 포함하고, 마스터 프로세스(100), 워커 프로세스(150) 및 파라미터 서버(200)는 원격 직접 메모리 접근(RDMA, Remote Direct Memory Access)을 지원하는 고속 네트워크로 연결된다.

[0045] 마스터 프로세스(100)는 파라미터 서버(200)에 원격 공유 메모리를 생성하고, 분산 딥러닝 프레임워크의 전체적인 제어를 담당한다. 그리고 마스터 프로세스(100)는 워커 프로세스들(150)에 원격 공유 메모리 정보를 전달하여, 워커 프로세스들(150)이 파라미터 서버(200) 상의 동일 메모리 영역에 접근할 수 있도록 한다. 반면, 워크 프로세스들(150)은 분산 트레이닝을 수행하고, 트레이닝한 결과를 저장한다.

[0046] 파라미터 서버(200)는 가중치(Weight) 파라미터 및 그래디언트(Gradient) 파라미터 중 적어도 어느 하나를 포함하는 분산 딥러닝 파라미터를 공유하기 위한 공유 메모리를 제공한다. 그리고 파라미터 서버(200)는 분산 딥러닝 프로세스들(100, 150)이 공유 메모리를 통해 공유한 딥러닝 파라미터를 이용하여, 분산 딥러닝 트레이닝을 수행하도록 한다.

[0048] 이하에서는 도 2를 통하여, 본 발명의 일실시예에 따른 파라미터 서버의 구성 및 기능에 대하여 더욱 상세하게 설명한다.

[0049] 도 2는 본 발명의 일실시예에 따른 파라미터 서버의 구성을 나타낸 블록도이다.

[0050] 도 2에 도시한 바와 같이, 파라미터 서버(200)는 통신 처리부(210), 공유 메모리 관리부(220), 파라미터 연산부(230) 및 이벤트 처리부(240)를 포함한다.

[0051] 먼저, 통신 처리부(210)는 마스터 프로세스 및 하나 이상의 워커 프로세스 중 적어도 어느 하나의 분산 딥러닝 트레이닝 엔진과 메시지를 송수신한다. 그리고 통신 처리부(210)는 마스터 프로세스 및 워커 프로세스 중 적어도 어느 하나의 원격 직접 메모리 접근(RDMA) 방식의 읽기 및 쓰기를 지원한다.

[0052] 그리고 공유 메모리 관리부(220)는 공유 메모리의 생성/할당 및 삭제/해제를 관리한다.

[0053] 공유 메모리 관리부(220)는 분산된 마스터 프로세스 또는 워커 프로세스로부터 수신한 파라미터용 원격 공유 메모리 생성 요청에 상응하도록 공유 메모리를 생성하고, 공유 메모리의 공유 메모리 생성키 및 접근 정보를 마스터 프로세스로 전송할 수 있다. 또한, 공유 메모리 관리부(220)는 워커 프로세스로부터 공유 메모리 할당 요청을 수신하고, 공유 메모리 할당 요청에 상응하도록 공유 메모리를 할당한다. 그리고 할당된 공유 메모리의 접근 정보를 워커 프로세스로 전송할 수 있다.

[0054] 그리고 공유 메모리 관리부(220)는 워커 프로세스로부터 공유 메모리 해제 요청을 수신하여 공유 메모리를 해제하고, 마스터 프로세스로부터 공유 메모리 삭제 요청을 수신하여 공유 메모리를 삭제할 수 있다.

[0055] 다음으로 파라미터 연산부(230)는 분산 딥러닝 파라미터를 계산한다. 이때, 분산 딥러닝 파라미터는 가중치 파라미터 및 그래디언트 파라미터를 포함할 수 있다.

[0056] 그리고 파라미터 연산부(230)는 두 개의 공유 메모리 영역에 대한 벡터/매트릭스 연산을 수행할 수 있으며, 벡터 연산은 제1 벡터(X)에 제1 상수(a)를 곱하는 scal 연산(X=aX), 제1 상수(a)를 곱한 제1 벡터(X)와 제2 벡터(Y)를 합하는 axpy 연산(Y=aX+Y), 제1 상수(a)를 곱한 제1 벡터(X)와 제2 상수(b)를 곱한 제2 벡터(Y)를 합하는 axpby 연산(Y=aX+bY) 등을 의미할 수 있다.

[0057] 또한, 파라미터 연산부(230)는 동기식으로 분산 딥러닝 파라미터를 공유하는 경우, 그래디언트 파라미터 누적을 연산하고, 마스터 프로세스의 마스터 가중치 파라미터를 입력받아 마스터 가중치 파라미터 영역을 업데이트할 수 있다.

그리고 비동기식으로 분산 딥러닝 파라미터를 공유하는 경우, 파라미터 연산부(230)는 공유 메모리의 마스터 가중치 파라미터 값으로 워커 프로세스의 워커 로컬 가중치 파라미터 영역을 업데이트하고, 분산 딥러닝을 수행한 워커 프로세스로부터 수신한 워커 그래디언트 파라미터로 마스터 가중치 파라미터 영역을 업데이트할 수 있다.

[0058] 마지막으로 이벤트 처리부(240)는 공유 메모리에 대한 이벤트 발생 시, 공유 메모리에 상응하는 마스터 프로세스 및 워커 프로세스 중 적어도 어느 하나로 이벤트의 발생을 알릴 수 있다. 이벤트 처리부(240)는 특정 공유 메모리 영역에 대한 이벤트를 해당 공유 메모리를 공유하는 지정된 분산 마스터 프로세스 또는 워커 프로세스에 알리는 통보 메시지를 전송할 수 있다.

[0059] 예를 들어, 특정 공유 메모리 영역이 업데이트 되었거나, 특정 공유 메모리 영역에 대해 설정된 연산이 완료된 경우, 이벤트 처리부(240)는 지정된 분산 딥러닝 트레이닝 엔진에 통보 메시지를 전송할 수 있다.

[0061] 이하에서는 도 3 내지 도 5를 통하여 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크의 동작 및 기능에 대하여 더욱 상세하게 설명한다.

[0062] 도 3은 본 발명의 일실시예에 따른 파라미터 공유를 위한 원격 공유 메모리 가상 매핑 매커니즘을 나타낸 예시도이다.

[0063] 도 3과 같이, 분산 딥러닝 트레이닝 엔진을 포함하는 마스터 프로세스(310) 및 워커 프로세스(320)는 파라미터를 공유하기 위하여, 파라미터 서버(330)에 원격 공유 메모리를 생성 및 할당하고, 로컬 물리 메모리(파라미터

의 임시 저장을 위한 호스트 물리 메모리 또는 GPU 등의 가속기 물리 메모리)를 할당하며, 가상 주소 공간에 맵핑한다.

[0064] 마스터 프로세스(310) 및 워커 프로세스(320) 각각은 분산 딥러닝 트레이닝 엔진 및 파라미터 서버 접근 지원부로 구성될 수 있으며, 분산 딥러닝 트레이닝 엔진은 딥러닝 모델 복사본(Model replica)을 이용하여 트레이닝을 수행할 수 있다. 이때, 분산 딥러닝 트레이닝 엔진의 역할은 마스터 프로세스(310)인지, 워커 프로세스(320)인지 여부에 따라 상이할 수 있다.

[0065] 마스터 프로세스(310)의 분산 딥러닝 트레이닝 엔진은 파라미터 서버(330)에 원격 공유 메모리를 생성하고, 하나 이상의 워커 프로세스(320)의 분산 딥러닝 트레이닝 엔진에 원격 공유 메모리의 정보를 전달하여, 워커 프로세스(320)들이 파라미터 서버(330) 상의 동일 메모리 영역에 접근할 수 있도록 한다. 이때, 원격 공유 메모리의 정보는 공유 메모리 생성키, 공유 메모리 크기 등을 포함할 수 있다.

[0066] 그리고 마스터 프로세스(310) 또는 워커 프로세스(320)의 분산 딥러닝 트레이닝 엔진은 파라미터 서버 접근 지원부를 통하여, 원격 계산 노드에서 실행된 파라미터 서버(330)를 이용할 수 있다. 이때, 파라미터 서버(330)가 원격 공유 메모리를 할당하면, 파라미터 서버 접근 지원부는 원격 공유 메모리와 동일한 크기의 로컬 물리 메모리를 할당받고, 할당받은 로컬 물리 메모리를 분산 딥러닝 트레이닝 엔진의 가상 주소 공간에 맵핑한다.

[0067] 마스터 프로세스(310) 또는 워커 프로세스(320)의 분산 딥러닝 트레이닝 엔진은 자신의 로컬 물리 메모리에 트레이닝된 파라미터를 저장하고, 명시적으로 파라미터 서버 접근 지원부가 제공하는 API를 이용하여 동기화(쓰기) 요청을 하면, 로컬 물리 메모리의 계산된 파라미터 데이터가 파라미터 서버(330)의 원격 공유 메모리로 복사된다. 또한, 마스터 프로세스(310) 또는 워커 프로세스(320)는 원격 공유 메모리의 업데이트된 파라미터를 읽어오는 동기화(읽기) 요청도 수행할 수 있다.

[0068] 설명의 편의를 위하여, 파라미터 서버(330)가 제공하는 메모리를 원격 공유 메모리로 명명하였으나, 이는 접근 방식이 공유 메모리 형태의 접근 방식인 것을 의미하며, 공유 메모리를 할당받은 프로세스간 자동 동기화 기능은 제공하지 않고, 원격 공유 메모리는 일종의 통신 버퍼로 활용됨을 의미할 수 있다.

[0070] 도 4는 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크의 기능을 설명하기 위한 구조도이다.

[0071] 도 4에 도시한 바와 같이, 분산 딥러닝 프레임워크는 분산 프로세스(410) 및 파라미터 서버(420)를 포함하며, 분산 프로세스(410)는 분산 딥러닝 트레이닝 엔진(411) 및 파라미터 서버 접근 지원부(415)를 포함할 수 있다.

[0072] 분산 딥러닝 트레이닝 엔진(411)의 관점에서, 파라미터 서버 접근 지원부(415)는 분산 프로세스(계산 노드)(410)에 함께 링크되어 라이브러리 형태로 제공될 수 있으며, 파라미터 서버 접근 지원부(415)의 기능은 모두 유저 레벨 라이브러리 형태로 구현될 수 있다. 또한, 파라미터 서버 접근 API만 라이브러리 형태로 구현되고, 이외의 기능은 장치 드라이버 형태로 구현될 수 있다.

[0073] 분산 딥러닝 트레이닝 엔진(411)은 분산 프로세스(410)에서 실행되며, 파라미터 서버 접근 지원부(415)에서 제공하는 파라미터 서버 접근 API를 이용하여 다른 분산 프로세스의 분산 딥러닝 트레이닝 엔진(411) 간 파라미터를 공유할 수 있다.

[0074] 파라미터 서버(420)는 별도의 프로세스에서 실행되며, 분산 프로세스(410)의 파라미터 서버 접근 지원부(415)와 Infiniband 등의 고속 네트워크 채널을 통해 메시지를 송수신하고, 원격 직접 메모리 접근(RDMA) 방식으로 원격 공유 메모리의 읽기/쓰기를 수행할 수 있다.

[0075] 분산 프로세스(410)의 분산 딥러닝 트레이닝 엔진(411)은 파라미터 서버 접근 지원부(415)의 파라미터 서버 접근 API(응용 프로그램 인터페이스)를 이용하여, 공유 메모리의 할당 및 해제, 명시적 공유 메모리 동기화(읽기/쓰기), 파라미터 연산 요청 등을 수행할 수 있다.

[0076] 그리고 파라미터 서버 접근 지원부(415)는 파라미터 서버 접근 API와 원격 공유 메모리 할당 요청 모듈, 공유 메모리 동기화 모듈, 공유 메모리 파라미터 연산 요청 모듈, 공유 메모리 이벤트 요청 모듈, 메시지 송수신 처리 모듈 및 고속 네트워크 통신 처리 모듈을 포함할 수 있다.

[0077] 그리고 파라미터 서버 접근 지원부(415)는 파라미터 서버 접근 API를 통해 분산 딥러닝 트레이닝 엔진(411)의 요청을 수신하면, 구성 모듈을 이용하여 요청에 상응하는 처리를 수행할 수 있다.

[0078] 예를 들어, 공유 메모리의 할당/해제 요청을 수신한 경우 파라미터 서버 접근 지원부(415)는 원격 공유 메모리 할당 요청 모듈을 이용하여 요청을 처리하고, 공유 메모리 동기화 요청을 수신하면 공유 메모리 동기화 모듈이

원격 메모리의 읽기 및 쓰기를 수행할 수 있다.

[0079] 그리고 파라미터 계산 요청을 수신한 경우 파라미터 서버 접근 지원부(415)는 공유 메모리 파라미터 연산 요청 모듈이 파라미터 서버(420)로 특정 공유 메모리 영역들 간의 연산을 요청할 수 있다. 또한, 이벤트 메시지의 송수신 요청을 수신하면, 파라미터 서버 접근 지원부(415)는 공유 메모리 이벤트 요청 모듈을 통하여 파라미터 서버로 이벤트 메시지의 전송을 요청할 수 있다.

[0080] 파라미터 서버(420)는 분산 프로세스(410)의 파라미터 서버 접근 지원부(415)의 요청을 처리하며, 원격 공유 메모리 할당 관리 모듈, 공유 메모리 파라미터 연산 모듈, 공유 메모리 이벤트 처리 모듈, 메시지 송수신 처리 모듈 및 고속 네트워크 통신 처리 모듈을 포함할 수 있다.

[0081] 원격 공유 메모리 할당 관리 모듈은 공유 메모리의 생성, 삭제, 할당 및 해제 요청을 처리하고, 공유 메모리 파라미터 연산 모듈은 2 개의 공유 메모리 영역에 대한 벡터/매트릭스 연산을 수행할 수 있다. 또한, 공유 메모리 이벤트 처리 모듈은 특정 공유 메모리 영역에 대한 이벤트를 해당 공유 메모리 영역을 생성 및 할당받은 분산 프로세스(410)의 분산 딥러닝 트레이닝 엔진(411)에 알리는 통보 메시지를 전송할 수 있다.

[0082] 설명의 편의를 위하여, 분산 프로세스(410)를 하나만 도시하였으나, 분산 딥러닝 프레임워크는 하나 이상의 분산 프로세스(410)를 포함할 수 있고, 분산 프로세스(410)는 분산 딥러닝 트레이닝 엔진(411)의 기능에 따라 마스터 프로세스 및 워커 프로세스로 구분될 수 있다.

[0084] 도 5는 본 발명의 일실시예에 따른 프로세스별 원격 공유 메모리 할당의 일 예를 나타낸 예시도이다.

[0085] 도 5와 같이, 마스터 프로세스(510)는 마스터 파라미터를 위한 원격 공유 메모리 생성을 담당한다. 그리고 마스터 프로세스(510)는 파라미터 서버(530)에 원격 공유 메모리를 생성하므로, 생성한 모든 원격 공유 메모리 영역에 접근할 수 있으며, 공유 메모리 생성 정보를 워커 프로세스(520)로 전송하여, 워커 프로세스(520)들이 마스터 영역에 접근할 수 있도록 한다.

[0086] 반면, 워커 프로세스(520)는 자신이 트레이닝한 결과를 저장하는 워커 그래디언트 파라미터 영역을 생성하고, 생성한 워커 그래디언트 파라미터 영역에 접근할 수 있다. 즉, 워커 프로세스(520)는 다른 워커 프로세스의 메모리 영역에 접근할 수 없으며, 마스터 파라미터 영역 및 해당 워커 프로세스(520)가 트레이닝한 결과를 저장하는 워커 파라미터 영역에만 접근 가능하다. 다시 말해, 제x 워커 프로세스(520_x)는 마스터 파라미터 영역 및 제x 워커 파라미터 영역에 접근 가능하다.

[0087] 설명의 편의를 위하여, 워커 프로세스(520)가 하나의 워커 파라미터 영역의 공유 메모리를 할당 받는 것으로 도시하였다. 그러나 실제 딥러닝 레이어별로 파라미터가 존재하므로, 워커 프로세스들(520)은 딥러닝 계층별로 하나의 마스터 파라미터와 워커 파라미터에 접근할 수 있으며, 도 5의 마스터 파라미터 영역 및 워커 파라미터 영역들은 다수의 공유 메모리 집합을 의미할 수 있다.

[0089] 이하에서는 도 6 내지 도 10을 통하여 본 발명의 일실시예에 따른 분산 딥러닝 파라미터 공유 방법에 대하여 더욱 상세하게 설명한다.

도 6은 본 발명의 일실시예에 따른 분산 딥러닝 파라미터 공유 방법을 나타낸 순서도이다.

먼저, 파라미터 서버(200)는 분산 딥러닝 프로세스의 요청에 상응하도록, 공유 메모리를 생성 및 할당한다(S110).

파라미터 서버(200)는 마스터 프로세스의 파라미터용 원격 공유 메모리 생성 요청에 상응하도록 공유 메모리를 생성하고, 워커 프로세스의 공유 메모리 할당 요청에 상응하도록 공유 메모리를 할당할 수 있다. 공유 메모리를 생성 및 할당하는 과정에 대해서는 후술할 도 7을 통하여 더욱 상세하게 설명한다.

그리고 파라미터 서버(200)는 공유 메모리의 마스터 가중치 파라미터 영역을 초기화하고(S120), 분산 딥러닝 프로세스들이 공유 메모리를 통해 공유한 딥러닝 파라미터를 이용하여, 분산 딥러닝 트레이닝을 수행하도록 한다(S130).

이때, 파라미터 서버(200)는 동기식 또는 비동기식으로 분산 딥러닝 파라미터를 공유하여, 분산 딥러닝 트레이닝을 수행하도록 할 수 있다. 파라미터 서버(200)가 동기식으로 분산 딥러닝 파라미터를 공유하는 과정에 대해서는 후술할 도 9를 통하여 더욱 상세하게 설명하고, 비동기식으로 분산 딥러닝 파라미터를 공유하는 과정에 대해서는 후술할 도 10을 통하여 더욱 상세하게 설명한다.

분산 딥러닝 트레이닝의 수행이 완료되면, 파라미터 서버(200)는 사용이 완료된 공유 메모리를 해제 및 삭제한다(S140).

파라미터 서버(200)는 워커 프로세스의 공유 메모리 해제 요청에 따라 공유 메모리를 해제하고, 마스터 프로세스로부터 공유 메모리 삭제 요청을 수신하면 공유 메모리를 삭제한다. 공유 메모리를 해제 및 삭제하는 과정에 대해서는 후술할 도 8을 통하여 더욱 상세하게 설명한다.

[0090] 도 7은 본 발명의 일실시예에 따른 원격 공유 메모리의 생성 및 할당 과정을 나타낸 순서도이다.

[0091] 먼저, 마스터 프로세스(100)는 파라미터 서버(200)로 원격 공유 메모리 생성 요청을 전송한다(S610).

[0092] 그리고 파라미터 서버(200)는 수신된 원격 공유 메모리 생성 요청에 상응하도록 공유 메모리를 생성하고(S620), 마스터 프로세스(100)로 공유 메모리 생성키 및 접근 정보를 전송한다(S630).

[0093] 이때, 파라미터 서버(200)는 생성된 공유 메모리에 접근하고자 하는 경우 필요한 정보인 공유 메모리 주소, 원격 메모리 접근키 등을 공유 메모리 생성키와 함께 마스터 프로세스(100)로 전송할 수 있다.

[0094] 다음으로 마스터 프로세스(100)는 파라미터 서버(200)로 공유 메모리 이벤트 설정 요청을 전송한다(S640).

[0095] 마스터 프로세스(100)는 업데이트 통지 이벤트, 누적 완료 이벤트 등의 공유 이벤트에 대한 이벤트 설정 요청을 파라미터 서버(200)로 전송할 수 있다. 여기서, 업데이트 통지 이벤트는 마스터 프로세스(100)가 특정 공유 메모리를 업데이트한 경우, 해당 공유 메모리를 공유하고 있는 모든 워커 프로세스들(150)로 알리는 메시지를 전송하도록 하는 이벤트를 의미한다.

[0096] 그리고 누적 완료 이벤트는 워커 프로세스들(150)이 특정 공유 메모리에 누적 수행을 완료한 경우, 마스터 프로세스(100)로 누적 완료를 알리는 메시지를 전송하는 이벤트를 의미한다.

[0097] 또한, 마스터 프로세스(100)는 공유 메모리 생성키를 하나 이상의 워커 프로세스(150)로 배포한다(S650).

[0098] 설명의 편의를 위하여 도 7에는 마스터 프로세스(100)가 하나의 워크 프로세스(150_1)로 공유 메모리 생성키를 배포하는 것으로 도시하였으나 이에 한정하지 않고, 마스터 프로세스(100)는 분산 딥러닝 프레임워크에 포함된 복수 개의 워커 프로세스(150)로 공유 메모리 생성키를 배포할 수 있다. 이때, 마스터 프로세스(100)는 마스터 프로세스(100)와 워커 프로세스(150)간 별도의 통신 채널을 이용하여 공유 메모리 생성키를 배포할 수 있다.

[0099] 그리고 공유 메모리 생성키를 수신한 제1 워커 프로세스(150_1)는 파라미터 서버(200)로 공유 메모리 할당 요청을 전송하고(S660), 파라미터 서버(200)는 공유 메모리를 할당한다(S670).

[0100] 마스터 프로세스(100)로부터 공유 메모리 생성키를 수신한 워커 프로세스(150)는 공유 메모리 생성키를 이용하여 파라미터 서버(200)로 공유 메모리 할당을 요청할 수 있다. 그리고 파라미터 서버(200)는 공유 메모리 생성키로 기 생성된 공유 메모리에 대한 할당을 수행할 수 있다.

[0101] 또한, 파라미터 서버(200)는 제1 워커 프로세스(150_1)로 할당된 공유 메모리 접근 정보를 전송한다(S680).

[0102] 파라미터 서버(200)는 공유 메모리 접근에 필요한 정보인 공유 메모리 주소, 원격 메모리 접근키 등의 공유 메모리 접근 정보를 워커 프로세스(150)로 전송한다. 그리고 공유 메모리 접근 정보를 수신한 워커 프로세스(150)는 공유 메모리 접근 정보를 이용하여 할당받은 원격 공유 메모리 주소에 RDMA 직접 읽기 또는 쓰기를 수행할 수 있다.

[0103] 그리고 분산 딥러닝 프레임워크에 포함된 모든 워커 프로세스들(150)이 S670 단계를 수행하여 공유 메모리 접근 정보를 수신하면, 마스터 프로세스(100)는 딥러닝 트레이닝을 수행할 수 있다.

[0104] 도 7과 같은 공유 메모리 할당 이외에, 워커 프로세스(150)가 주도적으로 공유 메모리를 할당하여, 다른 워커 프로세스들과 공유할 수 있으며, 마스터 프로세스(100) 및 워커 프로세스들(150)이 포함하는 딥러닝 트레이닝 엔진들 간의 공유 메모리 할당이 완료되면, 딥러닝 트레이닝 엔진들은 트레이닝을 시작할 수 있다. 그리고 딥러닝 트레이닝 중에는 마스터 프로세스(100)와 워커 프로세스(150)간에 다양한 형태로 딥러닝 파라미터가 공유될 수 있다.

[0106] 도 8은 본 발명의 일실시예에 따른 원격 공유 메모리의 삭제 및 해제 과정을 나타낸 순서도이다.

[0107] 제1 워커 프로세스(150_1)는 파라미터 서버(200)로 공유 메모리 해제 요청을 전송한다(S710).

[0108] 딥러닝 트레이닝이 완료된 후, 워커 프로세스(150) 각각은 자신이 할당받은 원격 공유 메모리의 해제를 파라미터 서버(200)에 요청할 수 있다.

[0109] 그리고 공유 메모리 해제 요청을 수신한 파라미터 서버(200)는 공유 메모리를 해제하고(S720), 제1 워커 프로세스(150_1)로 공유 메모리 해제를 통보한다(S730).

[0110] 여기서, 공유 메모리의 해제는 파라미터 서버(200)가 공유 메모리에 대한 공유 정보를 삭제하는 것을 의미할 수 있다.

[0111] 또한, 마스터 프로세스(100)는 파라미터 서버(200)로 원격 공유 메모리 삭제 요청을 전송하고(S740), 공유 메모리 삭제 요청을 수신한 파라미터 서버(200)는 공유 메모리를 삭제하며(S750), 마스터 프로세스(100)로 공유 메모리 삭제 완료를 통보한다(S760).

[0113] 이하에서는 도 9 및 도 10을 통하여 본 발명의 일실시예에 따른 분산 딥러닝 프레임워크 환경에서 동기식 및 비동기식으로 파라미터를 공유하는 방법에 대하여 더욱 상세하게 설명한다.

[0114] 파라미터 서버(200)에 원격 공유 메모리가 생성 및 할당된 후, 파라미터 서버(200)는 분산 딥러닝 프로세스들(100, 150)이 공유 메모리를 통해 딥러닝 파라미터를 공유하여, 분산 딥러닝 트레이닝을 수행하도록 할 수 있다. 즉, 마스터 프로세스(100) 및 하나 이상의 워커 프로세스(150)는 파라미터 서버(200)를 기반으로 딥러닝 파라미터를 공유하여, 딥러닝 트레이닝 과정을 반복 수행할 수 있다.

[0115] 여기서, 파라미터 서버(200)에 생성되는 파라미터는 마스터 가중치 파라미터(W_{Master}), 마스터 그래디언트 파라미터(G_{Master}) 및 워커x 그래디언트 파라미터(G_{worker_x})로 구분될 수 있다.

[0116] 그리고 딥러닝 트레이닝 과정에서 분산 딥러닝 파라미터는 도 9 또는 도 10의 과정을 통하여 동기식 또는 비동기식으로 공유될 수 있다. 이때, 도 9 및 도 10의 분산 딥러닝 파라미터를 공유하는 과정은 딥러닝 알고리즘에 따라 일부 순서의 변경 및 수정이 가능하다.

[0117] 또한, 도 9 및 도 10의 파라미터 공유 과정 각각은 도 7의 공유 메모리 생성 및 할당 과정을 수행한 후 수행될 수 있으며, 도 9 및 도 10의 과정을 수행한 후 도 8의 공유 메모리 삭제 및 해제 과정이 수행될 수 있다.

[0119] 도 9는 본 발명의 일실시예에 따른 동기식 파라미터 공유 방법을 설명하기 위한 순서도이다.

[0120] 먼저, 마스터 프로세스(100)는 파라미터 서버(200)의 마스터 가중치 파라미터 영역(W_{Master}) 및 마스터 그래디언트 파라미터 영역(G_{Master})을 초기화한다(S810).

[0121] 마스터 프로세스(100)는 마스터 프로세스(100)의 로컬 메모리에 초기화된 가중치 파라미터 값을 마스터 가중치 파라미터 영역에 쓰기하여, 마스터 가중치 파라미터 영역(W_{Master})을 초기화할 수 있다. 그리고 마스터 프로세스(100)는 모든 값을 0으로 설정하여 마스터 그래디언트 파라미터 영역(G_{Master})을 리셋할 수 있다.

[0122] 그리고 파라미터 서버(200)는 제1 워커 프로세스(150_1)로 마스터 가중치 파라미터(W_{Master})가 업데이트 되었음을 알린다(S820).

[0123] 파라미터 서버(200)는 마스터 가중치 파라미터(W_{Master})영역을 공유하는 하나 이상의 워커 프로세스들(150)로, 마스터 가중치 파라미터(W_{Master}) 영역이 업데이트 되었음을 알릴 수 있다.

[0124] 제1 워커 프로세스(150_1)는 마스터 가중치 파라미터(W_{Master})를 읽어와 워커 로컬 가중치 파라미터를 업데이트하고(S830), 딥러닝 트레이닝을 수행한다(S840).

[0125] 제1 워커 프로세스(150_1)는 워커 로컬 가중치 파라미터 영역을 공유 메모리의 마스터 가중치 파라미터의 값으로 업데이트할 수 있다. 즉, 워커 프로세스들(150)은 파라미터 서버(200)의 마스터 가중치 파라미터 영역을 RDMA 방식으로 읽어, 워커 로컬 가중치 파라미터(W_{Worker}) 영역으로 복사한다($W_{Worker} = W_{Master}$). 여기서, X는 워커 프로세스의 일련 번호를 의미하며, 제1 워커 프로세스(150_1)는 로컬 가중치 파라미터($W_{Worker1}$)을 업데이트할 수 있다.

[0126] 그리고 S840 단계에서 워커 프로세스들(150) 각각은 지정된 반복 트레이닝 횟수만큼 반복하여 딥러닝 트레이닝을 수행한다. 이때, 워커 프로세스들(150)은 가중치 파라미터는 업데이트하지 않고, 그래디언트 파라미터(G_{Worker})만 연산할 수도 있다.

[0127] 또한, 제1 워커 프로세스(150_1)는 파라미터 서버(200)에 워커 로컬 그래디언트 파라미터를 저장한다(S850).

[0128] 딥러닝 트레이닝을 수행한 워커 프로세스들(150)은 학습된 워커 로컬 그래디언트 파라미터(G_{Worker})를 공유 메모리의 워커 그래디언트 파라미터 영역에 RDMA 쓰기 한다. 즉, 제1 워커 프로세스(150_1)는 제1 워커 로컬 그래디언트 파라미터($G_{Worker1}$)를 제1 워커 파라미터 영역에 RDMA 쓰기 할 수 있다.

[0129] 그리고 제1 워커 프로세스(150_1)는 파라미터 서버(200)로 그래디언트 파라미터 누적 연산을 요청하고(S860), 파라미터 서버(200)는 요청된 그래디언트 파라미터 영역들 간의 그래디언트 파라미터 누적 연산을 수행한다(S870).

[0130] 제1 워커 프로세스(150_1)는 공유 메모리의 제1 워커 파라미터 영역에 저장된 워커 로컬 그래디언트 파라미터(G_{Worker})를 마스터 그래디언트 파라미터(G_{Master})에 누적하도록, 파라미터 서버(200)에 요청한다. 그리고 파라미터 서버(200)는 요청된 그래디언트 파라미터 영역들 간의 파라미터를 누적하는 연산인 $G_{Master}' = G_{Master} + G_{Worker}$ 연산을 수행할 수 있다.

[0131] 모든 워커 프로세스(150)들의 그래디언트 파라미터 누적 연산이 완료되면 파라미터 서버(200)는 마스터 프로세스(100)로 그래디언트 파라미터(G_{Master}) 누적 연산의 완료를 통보한다(S880).

[0132] 마스터 프로세스(100)는 분산 딥러닝 프레임워크에 포함된 모든 워커 프로세스들(150)의 그래디언트 파라미터 누적이 완료될 때까지 대기한 후, 파라미터 서버(200)로부터 누적 완료된 마스터 그래디언트 파라미터(G_{Master}) 영역을 읽어온다(S890).

[0133] 이때, 마스터 프로세스(100)는 모든 워커 프로세스들(150)의 그래디언트 파라미터가 누적된 마스터 그래디언트 파라미터 영역(G_{Master})을 RDMA 방식으로 읽어올 수 있다.

[0134] 그리고 마스터 프로세스(100)는 마스터 가중치 파라미터(W_{Master}')를 연산하며(S900), 마스터 가중치 파라미터(W_{Master}')를 파라미터 서버(200)에 업데이트한다(S910).

[0135] 마스터 프로세스(100)는 S890 단계에서 읽어온 그래디언트 누적 값(G_{Master})의 평균을 이용하여, 마스터 가중치 파라미터(W_{Master}')를 연산할 수 있다. 또한, 마스터 프로세스(100)는 새로 업데이트된 마스터 가중치 파라미터(W_{Master}')를 파라미터 서버(200)의 마스터 가중치 파라미터 영역에 저장할 수 있다.

[0136] 마스터 프로세스(100) 및 워커 프로세스들(150)은 지정된 반복 트레이닝 횟수만큼 S820 단계 내지 S910 단계의 수행을 반복할 수 있다.

[0138] 도 10은 본 발명의 일실시예에 따른 비동기식 파라미터 공유 방법을 설명하기 위한 순서도이다.

[0139] 먼저, 마스터 프로세스(100)는 파라미터 서버(200)의 마스터 가중치 파라미터(W_{Master}) 영역을 초기화한다(S1010). 그리고 파라미터 서버(200)는 제1 워커 프로세스(150_1)로 마스터 가중치 파라미터(W_{Master})의 업데이트를 통보한다(S1020).

[0140] 설명의 편의를 위하여, 파라미터 서버(200)가 제1 워커 프로세스(150_1)로 마스터 가중치 파라미터의 업데이트를 통보하는 것으로 설명하였으나 이에 한정하지 않고, 파라미터 서버(200)는 분산 딥러닝 프레임워크에 포함된 하나 이상의 워커 프로세스들(150)로 마스터 가중치 파라미터(W_{Master})가 업데이트 되었음을 알릴 수 있다.

[0141] 다음으로, 제1 워커 프로세스(150_1)는 공유 메모리의 마스터 가중치 파라미터(W_{Master})를 읽어와, 워커 로컬 가중치 파라미터(W_{Worker}) 영역을 업데이트하고(S1030), 딥러닝 트레이닝을 수행한다(S1040).

[0142] 제1 워커 프로세스(150_1)는 RDMA 방식으로 마스터 가중치 파라미터(W_{Master})를 읽어올 수 있으며, 읽어온 마스터 파라미터(W_{Master})를 워커 로컬 가중치 파라미터(W_{Worker})로 복사($W_{Worker} = W_{Master}$)하여, 워커 로컬 가중치 파라미터

(W_{Worker})를 업데이트할 수 있다. 그리고 제1 워커 프로세스(150_1)는 지정된 반복 횟수만큼 딥러닝 트레이닝을 수행하여, 워커 로컬 그래디언트 파라미터(G_{Worker})를 계산할 수 있다.

[0143] 딥러닝 트레이닝을 수행한 제1 워커 프로세스(150_1)는 새로 학습된 워커 그래디언트 파라미터(G_{Worker})를 공유 메모리에 RDMA 쓰기하여 업데이트한다(S1050). 그리고 제1 워커 프로세스(150_1)는 마스터 파라미터 서버(200)에 마스터 가중치 파라미터(W_{Master})의 업데이트를 요청한다(S1060).

[0144] 파라미터 서버(200)는 마스터 가중치 파라미터(W_{Master})의 업데이트를 수행하고(S1070), 업데이트를 요청한 제1 워커 프로세스(150_1)로 업데이트의 완료를 통보한다(S1080).

[0145] 이때, 파라미터 서버(200)는 복수의 워커 프로세스들(150)로부터 수신된 마스터 가중치 파라미터의 업데이트 수행 요청을 동시에 수행하지 않고, 순차적으로 처리할 수 있다.

[0146] 그리고 파라미터 서버(200)는 마스터 파라미터 영역의 업데이트가 완료되었음을 하나 이상의 워커 프로세스(150)에 통보할 수 있다. 이때, 딥러닝 트레이닝이 종료되지 않은 경우, S1030 단계 내지 S1080 단계의 과정을 반복하여 수행할 수 있다.

[0147] 도 9 및 도 10에는 도시하지 않았으나, 딥러닝 트레이닝 종료 시, 마스터 가중치 파라미터를 저장하는 과정을 수행한 후, 딥러닝 트레이닝을 종료할 수 있다.

[0149] 이상에서와 같이 본 발명에 따른 파라미터 서버 및 그것에 의해 수행되는 분산 딥러닝 파라미터 공유 방법은 상기한 바와 같이 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상기 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

부호의 설명

[0150] 100: 마스터 프로세스 150: 워커 프로세스

200: 파라미터 서버 210: 통신 처리부

220: 공유 메모리 관리부 230: 파라미터 연산부

240: 이벤트 처리부 310: 마스터 프로세스

320: 워커 프로세스 330: 파라미터 서버

410: 분산 프로세스 411: 분산 딥러닝 트레이닝 엔진

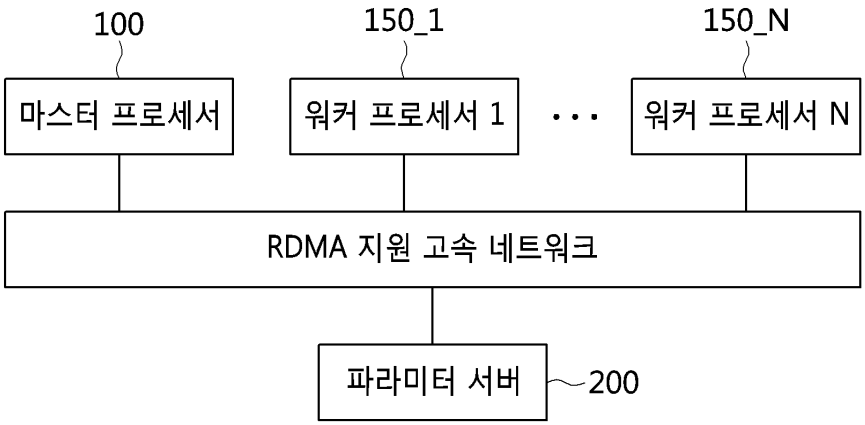
415: 파라미터 서버 접근 지원부 420: 파라미터 서버

510: 마스터 프로세스 520: 워커 프로세스

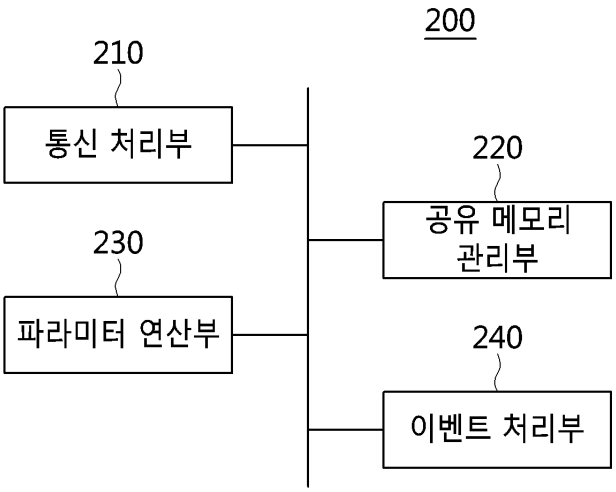
530: 파라미터 서버

도면

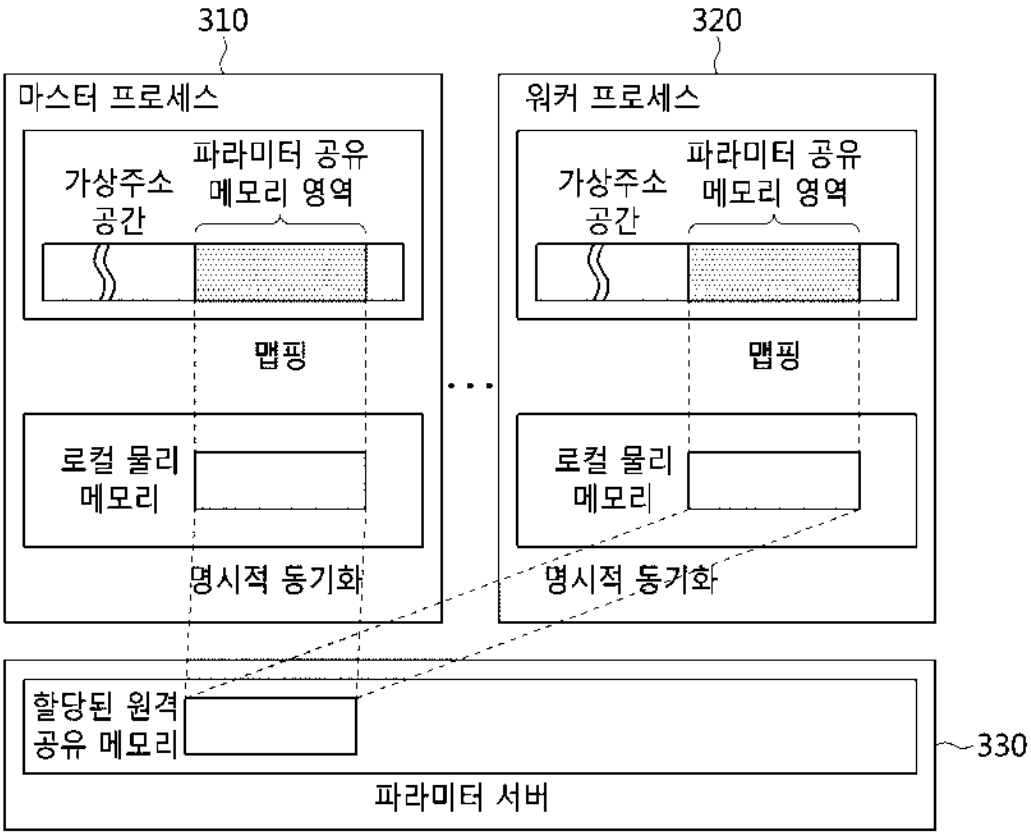
도면1


100
150_1
150_N
마스터 프로세서
워커 프로세서 1
• • •
워커 프로세서 N
RDMA 지원 고속 네트워크
파라미터 서버
200

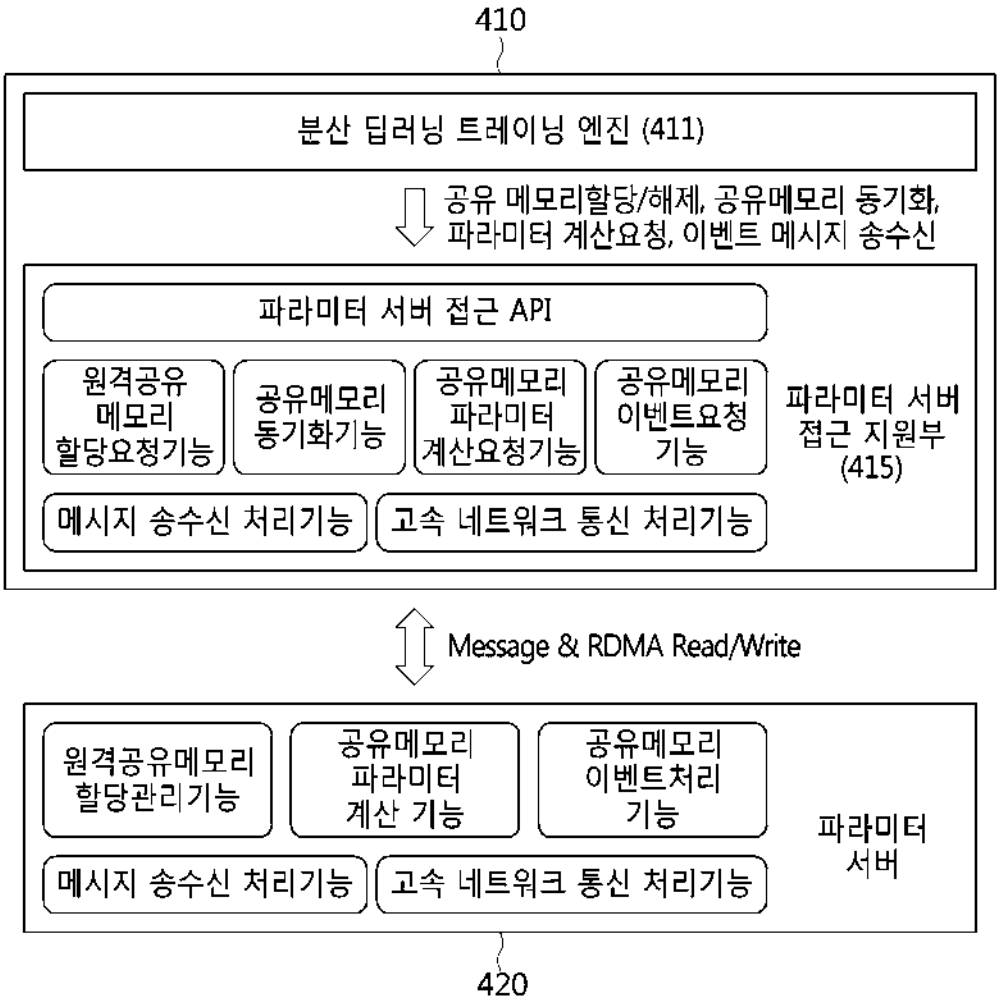
도면2


200
210
통신 처리부
220
공유 메모리 관리부
230
파라미터 연산부
240
이벤트 처리부

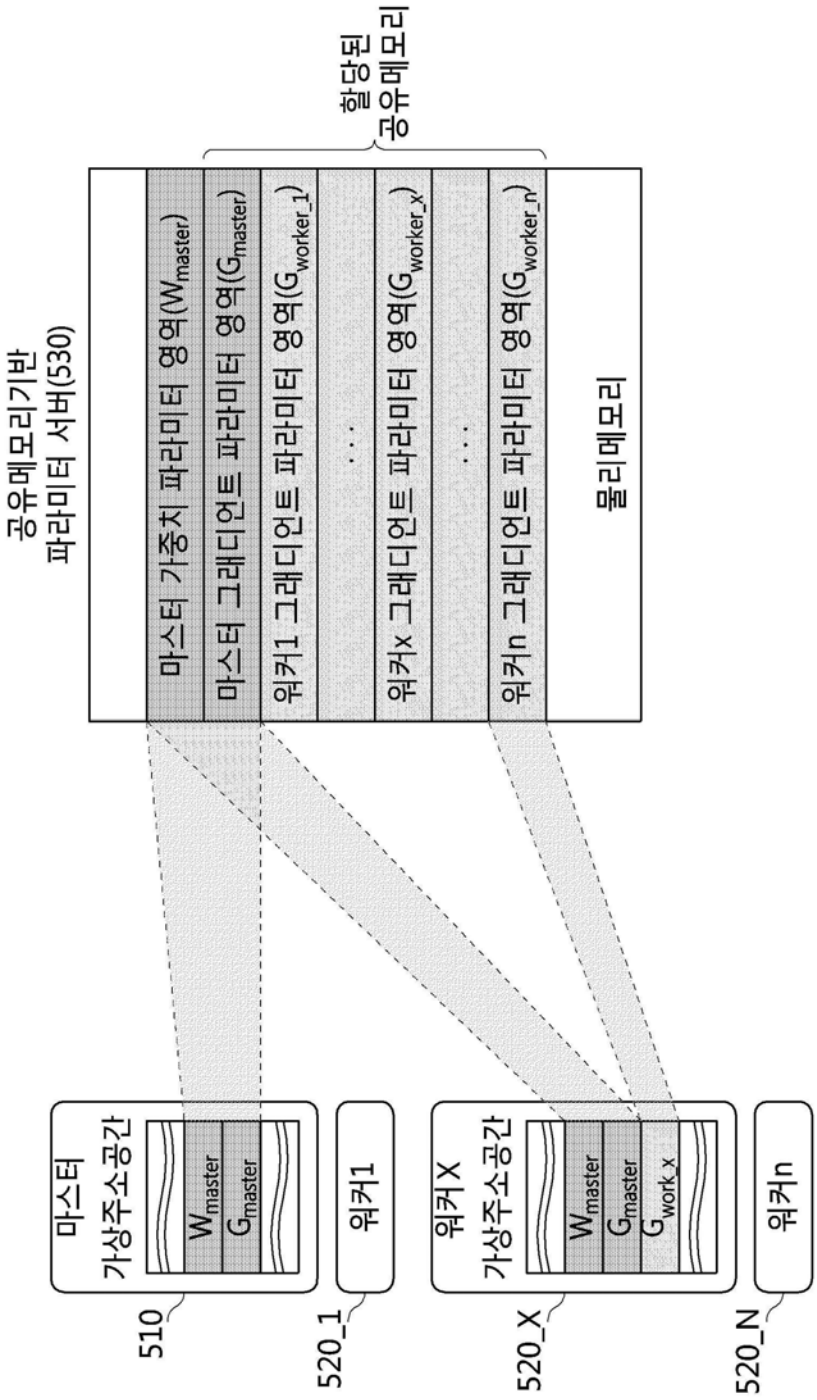
도면3


310
320
마스터 프로세스
워커 프로세스
가상주소 공간
파라미터 공유 메모리 영역
맵핑
로컬 물리 메모리
명시적 동기화
할당된 원격 공유 메모리
330
파라미터 서버

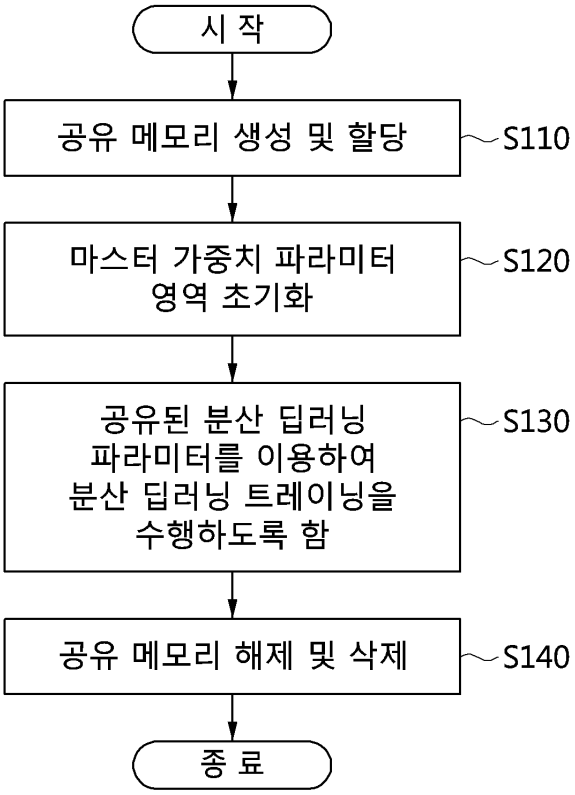
도면4


410
분산 딥러닝 트레이닝 엔진 (411)
공유 메모리할당/해제, 공유메모리 동기화,
파라미터 계산요청, 이벤트 메시지 송수신
파라미터 서버 접근 API
원격공유 메모리 할당요청기능
공유메모리 동기화기능
공유메모리 파라미터 계산요청기능
공유메모리 이벤트요청 기능
파라미터 서버 접근 지원부 (415)
메시지 송수신 처리기능
고속 네트워크 통신 처리기능
Message & RDMA Read/Write
원격공유메모리 할당관리기능
공유메모리 파라미터 계산 기능
공유메모리 이벤트처리 기능
파라미터 서버
메시지 송수신 처리기능
고속 네트워크 통신 처리기능
420

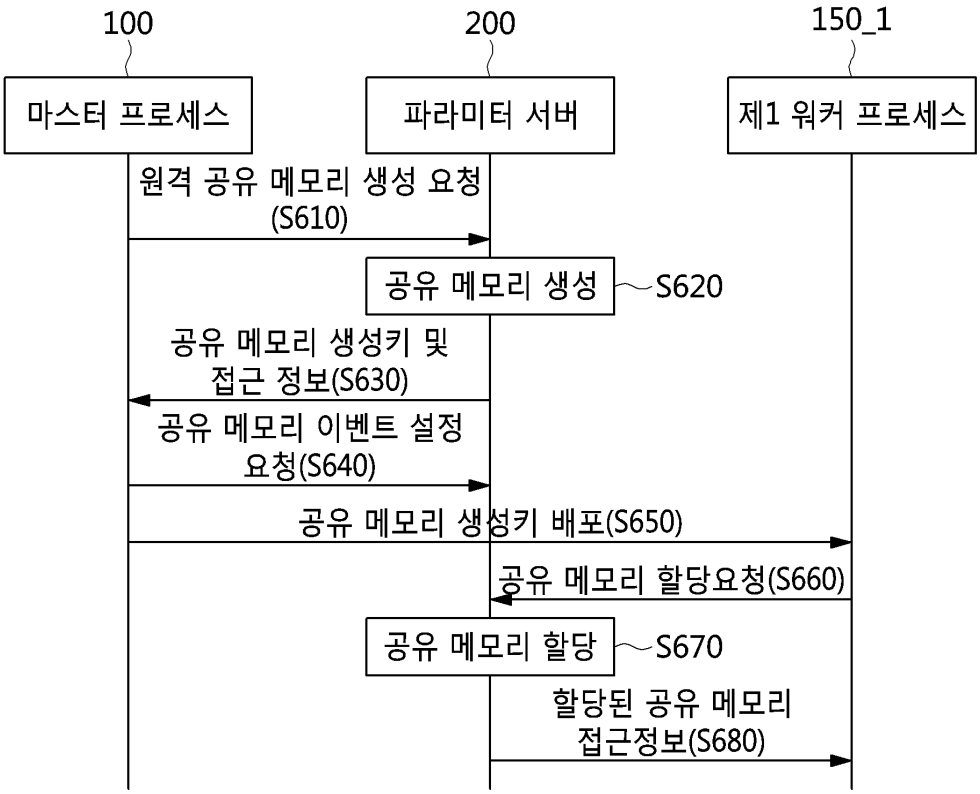
도면5


할당된
공유메모리
공유메모리기반
파라미터 서버(530)
마스터 가중치 파라미터 영역(W_{master})
마스터 그래디언트 파라미터 영역(G_{master})
워커1 그래디언트 파라미터 영역(G_{worker_1})
...
워커x 그래디언트 파라미터 영역(G_{worker_x})
...
워커n 그래디언트 파라미터 영역(G_{worker_n})
물리메모리
마스터
가상주소공간
W_{master}
G_{master}
510
520_1
워커1
워커 X
가상주소공간
W_{master}
G_{master}
G_{work_x}
520_X
520_N
워커n

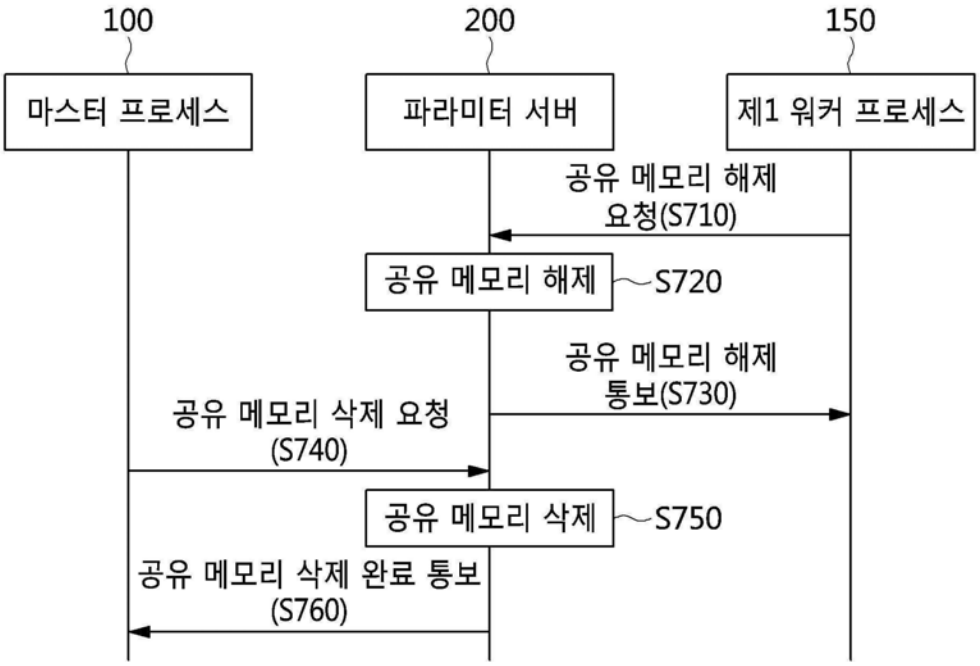
도면6


시 작
공유 메모리 생성 및 할당
S110
마스터 가중치 파라미터 영역 초기화
S120
공유된 분산 딥러닝 파라미터를 이용하여 분산 딥러닝 트레이닝을 수행하도록 함
S130
공유 메모리 해제 및 삭제
S140
종 료

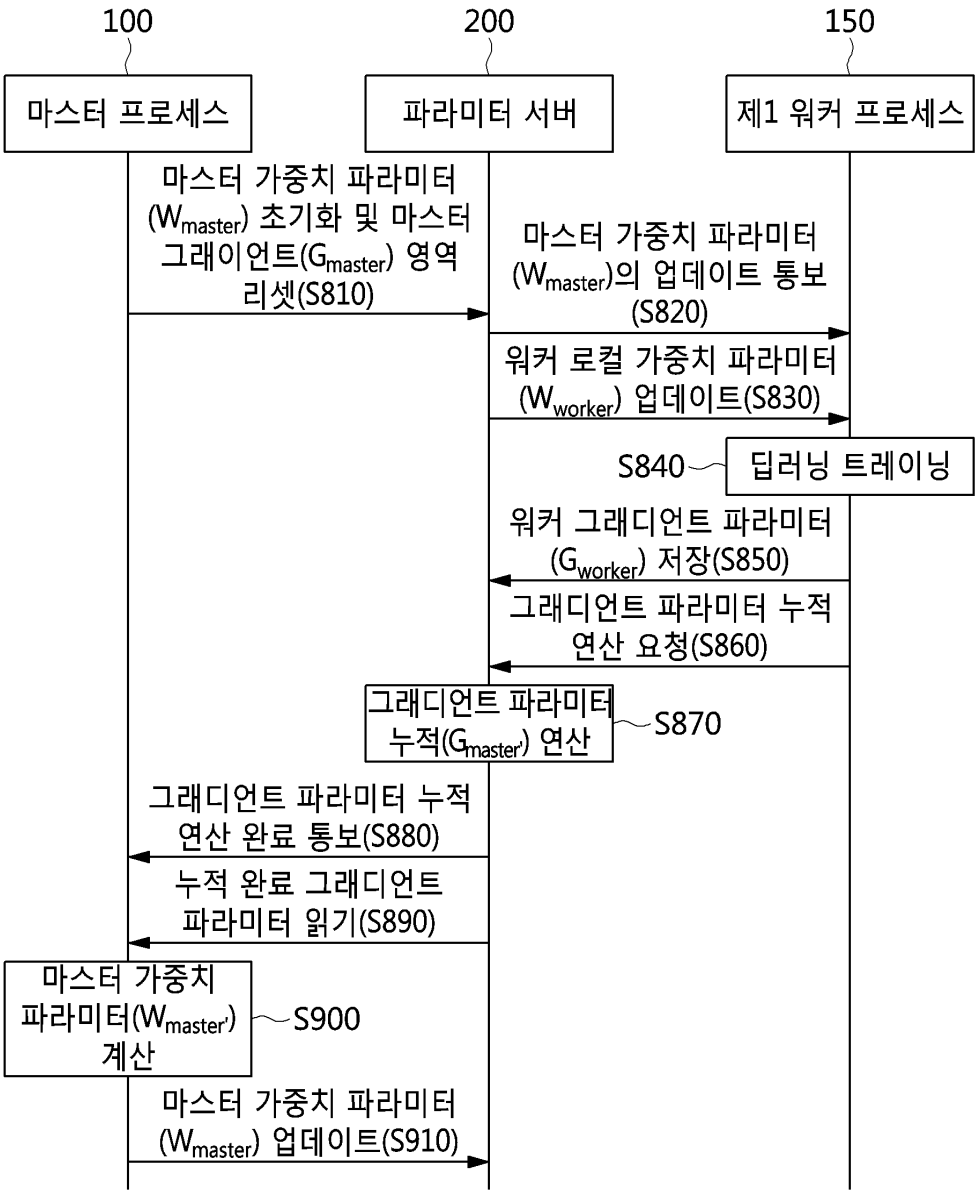
도면7


100
200
150_1
마스터 프로세스
파라미터 서버
제1 워커 프로세스
원격 공유 메모리 생성 요청 (S610)
공유 메모리 생성
S620
공유 메모리 생성키 및 접근 정보(S630)
공유 메모리 이벤트 설정 요청(S640)
공유 메모리 생성키 배포(S650)
공유 메모리 할당요청(S660)
공유 메모리 할당
S670
할당된 공유 메모리 접근정보(S680)

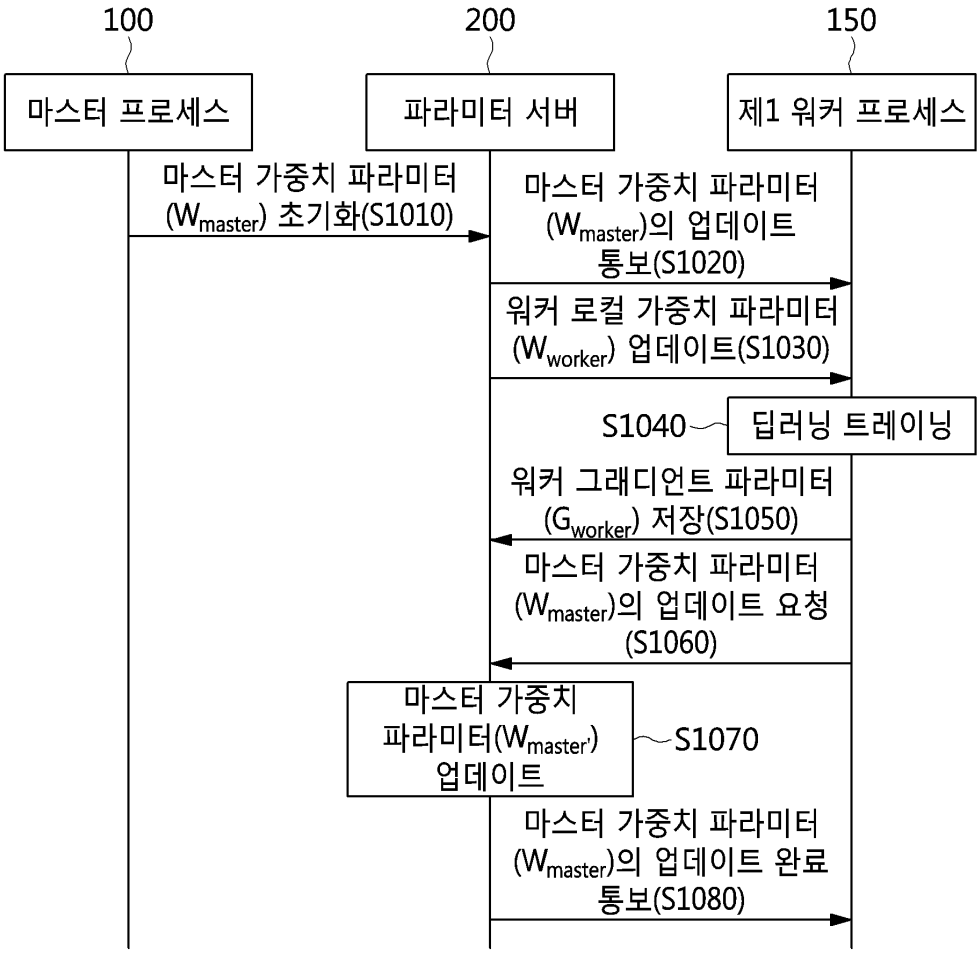
도면8


100
200
150
마스터 프로세스
파라미터 서버
제1 워커 프로세스
공유 메모리 해제 요청(S710)
공유 메모리 해제
S720
공유 메모리 해제 통보(S730)
공유 메모리 삭제 요청 (S740)
공유 메모리 삭제
S750
공유 메모리 삭제 완료 통보 (S760)

도면9


100
200
150
마스터 프로세스
파라미터 서버
제1 워커 프로세스
마스터 가중치 파라미터 (W_{master}) 초기화 및 마스터 그래이언트(G_{master}) 영역 리셋(S810)
마스터 가중치 파라미터 (W_{master})의 업데이트 통보 (S820)
워커 로컬 가중치 파라미터 (W_{worker}) 업데이트(S830)
S840
딥러닝 트레이닝
워커 그래디언트 파라미터 (G_{worker}) 저장(S850)
그래디언트 파라미터 누적 연산 요청(S860)
그래디언트 파라미터 누적($G_{master'}$) 연산
S870
그래디언트 파라미터 누적 연산 완료 통보(S880)
누적 완료 그래디언트 파라미터 읽기(S890)
마스터 가중치 파라미터($W_{master'}$) 계산
S900
마스터 가중치 파라미터 (W_{master}) 업데이트(S910)

도면10



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 1

【변경전】

파라미터 서버에 의해 수행되는 분산 딥러닝 파라미터 공유 방법에 있어서,

마스터 프로세스 및 적어도 하나의 워커 프로세스를 포함하는 분산 딥러닝 프로세스의 요청에 상응하도록, 원격 공유 메모리를 생성 및 할당하는 단계,

상기 원격 공유 메모리의 마스터 가중치 파라미터 영역을 초기화하는 단계,

상기 분산 딥러닝 프로세스들이 상기 원격 공유 메모리를 통해 공유한 분산 딥러닝 파라미터를 이용하여 분산 딥러닝 트레이닝을 수행하는 단계, 그리고

상기 분산 딥러닝 트레이닝의 수행이 완료된 후, 사용이 완료된 상기 원격 공유 메모리를 해제 및 삭제하는 단계를 포함하고,

상기 원격 공유 메모리를 생성 및 할당하는 단계는,

상기 마스터 프로세스의 요청에 따라 원격 공유 메모리를 생성하고, 상기 적어도 하나의 워커 프로세스에게 상기 원격 공유 메모리에 접근하기 위한 접근 정보를 전달하고,

상기 마스터 프로세스와 상기 적어도 하나의 워커 프로세스가, 상기 원격 공유 메모리에 상응하는 각각의 로컬 물리 메모리를 할당하고, 상기 각각의 로컬 물리 메모리를 분산 딥러닝 트레이닝 엔진의 가상 주소 공간에 맵핑

하고,

상기 분산 딥러닝 트레이닝을 수행하는 단계는

상기 파라미터 서버, 상기 마스터 프로세스 및 상기 적어도 하나의 워커 프로세스가, 상기 각각의 로컬 물리 메모리와 상기 원격 공유 메모리의 명시적 동기화를 통해 상기 분산 딥러닝 파라미터를 공유하여 상기 분산 딥러닝 트레이닝을 수행하고,

상기 원격 공유 메모리는

상기 마스터 프로세스에 의해 생성되고, 마스터 가중치 파라미터와 마스터 그래디언트 파라미터를 저장하는 마스터 영역; 및

상기 적어도 하나의 워커 프로세스의 개수에 상응하도록 생성되고, 적어도 하나의 워커 그래디언트 파라미터를 각각 저장하는 적어도 하나의 워커 영역;

을 포함하고,

상기 분산 딥러닝 트레이닝을 수행하는 단계는

상기 마스터 프로세스가, 상기 마스터 영역에 상기 마스터 가중치 파라미터와 상기 마스터 그래디언트 파라미터를 업데이트하고, 상기 적어도 하나의 워커 프로세스에게 상기 마스터 영역에 접근하기 위한 접근 정보를 전송하고,

상기 적어도 하나의 워커 프로세스가, 상기 접근 정보를 이용하여 상기 마스터 영역에 접근하여 상기 마스터 가중치 파라미터를 자신의 워커 가중치 파라미터로 업데이트하고,

상기 원격 공유 메모리의 적어도 하나의 워커 영역 중 자신이 생성한 워커 영역에 상기 분산 딥러닝 트레이닝을 수행한 결과로부터 학습된 워커 그래디언트 파라미터를 업데이트하고,

상기 파라미터 서버가, 상기 적어도 하나의 워커 프로세스로부터 상기 적어도 하나의 워커 영역에 상기 적어도 워커 그래디언트 파라미터가 업데이트되었음을 알림받으면, 상기 적어도 워커 그래디언트 파라미터를 상기 마스터 그래디언트 파라미터에 업데이트하고,

상기 마스터 서버가, 업데이트된 상기 마스터 그래디언트 파라미터를 이용하여 상기 마스터 가중치 파라미터를 업데이트하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

【변경후】

파라미터 서버에 의해 수행되는 분산 딥러닝 파라미터 공유 방법에 있어서,

마스터 프로세스 및 적어도 하나의 워커 프로세스를 포함하는 분산 딥러닝 프로세스의 요청에 상응하도록, 원격 공유 메모리를 생성 및 할당하는 단계,

상기 원격 공유 메모리의 마스터 가중치 파라미터 영역을 초기화하는 단계,

상기 분산 딥러닝 프로세스들이 상기 원격 공유 메모리를 통해 공유한 분산 딥러닝 파라미터를 이용하여 분산 딥러닝 트레이닝을 수행하는 단계, 그리고

상기 분산 딥러닝 트레이닝의 수행이 완료된 후, 사용이 완료된 상기 원격 공유 메모리를 해제 및 삭제하는 단계를 포함하고,

상기 원격 공유 메모리를 생성 및 할당하는 단계는,

상기 마스터 프로세스의 요청에 따라 원격 공유 메모리를 생성하고, 상기 적어도 하나의 워커 프로세스에게 상기 원격 공유 메모리에 접근하기 위한 접근 정보를 전달하고,

상기 마스터 프로세스와 상기 적어도 하나의 워커 프로세스가, 상기 원격 공유 메모리에 상응하는 각각의 로컬 물리 메모리를 할당하고, 상기 각각의 로컬 물리 메모리를 분산 딥러닝 트레이닝 엔진의 가상 주소 공간에 맵핑하고,

상기 분산 딥러닝 트레이닝을 수행하는 단계는

상기 파라미터 서버, 상기 마스터 프로세스 및 상기 적어도 하나의 워커 프로세스가, 상기 각각의 로컬 물리 메모리와 상기 원격 공유 메모리의 명시적 동기화를 통해 상기 분산 딥러닝 파라미터를 공유하여 상기 분산 딥러

닝 트레이닝을 수행하고,

상기 원격 공유 메모리는

상기 마스터 프로세스에 의해 생성되고, 마스터 가중치 파라미터와 마스터 그래디언트 파라미터를 저장하는 마스터 영역; 및

상기 적어도 하나의 워커 프로세스의 개수에 상응하도록 생성되고, 적어도 하나의 워커 그래디언트 파라미터를 각각 저장하는 적어도 하나의 워커 영역;

을 포함하고,

상기 분산 딥러닝 트레이닝을 수행하는 단계는

상기 마스터 프로세스가, 상기 마스터 영역에 상기 마스터 가중치 파라미터와 상기 마스터 그래디언트 파라미터를 업데이트하고, 상기 적어도 하나의 워커 프로세스에게 상기 마스터 영역에 접근하기 위한 접근 정보를 전송하고,

상기 적어도 하나의 워커 프로세스가, 상기 접근 정보를 이용하여 상기 마스터 영역에 접근하여 상기 마스터 가중치 파라미터를 자신의 워커 가중치 파라미터로 업데이트하고,

상기 원격 공유 메모리의 적어도 하나의 워커 영역 중 자신이 생성한 워커 영역에 상기 분산 딥러닝 트레이닝을 수행한 결과로부터 학습된 워커 그래디언트 파라미터를 업데이트하고,

상기 파라미터 서버가, 상기 적어도 하나의 워커 프로세스로부터 상기 적어도 하나의 워커 영역에 상기 적어도 하나의 워커 그래디언트 파라미터가 업데이트되었음을 알림받으면, 상기 적어도 하나의 워커 그래디언트 파라미터를 상기 마스터 그래디언트 파라미터에 업데이트하고,

상기 마스터 프로세스가, 업데이트된 상기 마스터 그래디언트 파라미터를 이용하여 상기 마스터 가중치 파라미터를 업데이트하는 것을 특징으로 하는 분산 딥러닝 파라미터 공유 방법.

【직권보정 2】

【보정항목】 청구범위

【보정세부항목】 청구항 8

【변경전】

마스터 프로세스 및 적어도 하나의 워커 프로세스를 포함하는 분산 딥러닝 프로세스의 요청과 관련된 메시지를 송수신하는 통신 처리부,

상기 분산 딥러닝 프로세스의 요청에 상응하도록, 분산 딥러닝 파라미터를 저장하기 위한 원격 공유 메모리를 생성, 할당 및 해제하는 원격 공유 메모리 관리부, 그리고

상기 분산 딥러닝 프로세스가 상기 원격 공유 메모리를 통해 공유한 분산 딥러닝 파라미터를 이용하여 분산 딥러닝 트레이닝을 수행하는 파라미터 연산부를 포함하고,

상기 원격 공유 메모리 관리부는

상기 마스터 프로세스의 요청에 따라 상기 원격 공유 메모리를 생성하고, 상기 적어도 하나의 워커 프로세스에게 상기 원격 공유 메모리에 접근하기 위한 접근 정보를 전달하고,

상기 마스터 프로세스와 상기 적어도 하나의 워커 프로세스는

상기 원격 공유 메모리에 상응하는 각각의 로컬 물리 메모리를 할당하고, 상기 각각의 로컬 물리 메모리를 분산 딥러닝 트레이닝 엔진의 가상 주소 공간에 맵핑하고,

상기 파라미터 연산부는

상기 마스터 프로세스 및 상기 적어도 하나의 워커 프로세스와 함께, 상기 각각의 로컬 물리 메모리와 상기 원격 공유 메모리의 명시적 동기화를 통해 상기 분산 딥러닝 파라미터를 공유하여 상기 분산 딥러닝 트레이닝을 수행하고,

상기 원격 공유 메모리는

상기 마스터 프로세스에 의해 생성되고, 마스터 가중치 파라미터와 마스터 그래디언트 파라미터를 저장하는 마스터 영역; 및

상기 적어도 하나의 워커 프로세스의 개수에 상응하도록 생성되고, 적어도 하나의 워커 그래디언트 파라미터를 각각 저장하는 적어도 하나의 워커 영역;

을 포함하고,

상기 마스터 프로세스는

상기 마스터 영역에 상기 마스터 가중치 파라미터와 상기 마스터 그래디언트 파라미터를 업데이트하고,

상기 적어도 하나의 워커 프로세스에게 상기 마스터 영역에 접근하기 위한 접근 정보를 전송하고,

상기 적어도 하나의 워커 프로세스는

상기 접근 정보를 이용하여 상기 마스터 영역에 접근하여 상기 마스터 가중치 파라미터를 자신의 워커 가중치 파라미터로 업데이트하고,

상기 원격 공유 메모리의 적어도 하나의 워커 영역 중 자신이 생성한 워커 영역에 상기 분산 딥러닝 트레이닝을 수행한 결과로부터 학습된 워커 그래디언트 파라미터를 업데이트하고,

상기 파라미터 연산부는

상기 적어도 하나의 워커 프로세스로부터 상기 적어도 하나의 워커 영역에 상기 적어도 워커 그래디언트 파라미터가 업데이트되었음을 알림받으면, 상기 적어도 워커 그래디언트 파라미터를 상기 마스터 그래디언트 파라미터에 누적 연산하여 업데이트하고,

상기 마스터 서버는

업데이트된 상기 마스터 그래디언트 파라미터를 이용하여 상기 마스터 가중치 파라미터를 업데이트하는 것을 특징으로 하는 파라미터 서버.

【변경후】

마스터 프로세스 및 적어도 하나의 워커 프로세스를 포함하는 분산 딥러닝 프로세스의 요청과 관련된 메시지를 송수신하는 통신 처리부,

상기 분산 딥러닝 프로세스의 요청에 상응하도록, 분산 딥러닝 파라미터를 저장하기 위한 원격 공유 메모리를 생성, 할당 및 해제하는 원격 공유 메모리 관리부, 그리고

상기 분산 딥러닝 프로세스가 상기 원격 공유 메모리를 통해 공유한 분산 딥러닝 파라미터를 이용하여 분산 딥러닝 트레이닝을 수행하는 파라미터 연산부를 포함하고,

상기 원격 공유 메모리 관리부는

상기 마스터 프로세스의 요청에 따라 상기 원격 공유 메모리를 생성하고, 상기 적어도 하나의 워커 프로세스에게 상기 원격 공유 메모리에 접근하기 위한 접근 정보를 전달하고,

상기 마스터 프로세스와 상기 적어도 하나의 워커 프로세스는

상기 원격 공유 메모리에 상응하는 각각의 로컬 물리 메모리를 할당하고, 상기 각각의 로컬 물리 메모리를 분산 딥러닝 트레이닝 엔진의 가상 주소 공간에 맵핑하고,

상기 파라미터 연산부는

상기 마스터 프로세스 및 상기 적어도 하나의 워커 프로세스와 함께, 상기 각각의 로컬 물리 메모리와 상기 원격 공유 메모리의 명시적 동기화를 통해 상기 분산 딥러닝 파라미터를 공유하여 상기 분산 딥러닝 트레이닝을 수행하고,

상기 원격 공유 메모리는

상기 마스터 프로세스에 의해 생성되고, 마스터 가중치 파라미터와 마스터 그래디언트 파라미터를 저장하는 마스터 영역; 및

상기 적어도 하나의 워커 프로세스의 개수에 상응하도록 생성되고, 적어도 하나의 워커 그래디언트 파라미터를

각각 저장하는 적어도 하나의 워커 영역;

을 포함하고,

상기 마스터 프로세스는

상기 마스터 영역에 상기 마스터 가중치 파라미터와 상기 마스터 그래디언트 파라미터를 업데이트하고,

상기 적어도 하나의 워커 프로세스에게 상기 마스터 영역에 접근하기 위한 접근 정보를 전송하고,

상기 적어도 하나의 워커 프로세스는

상기 접근 정보를 이용하여 상기 마스터 영역에 접근하여 상기 마스터 가중치 파라미터를 자신의 워커 가중치 파라미터로 업데이트하고,

상기 원격 공유 메모리의 적어도 하나의 워커 영역 중 자신이 생성한 워커 영역에 상기 분산 딥러닝 트레이닝을 수행한 결과로부터 학습된 워커 그래디언트 파라미터를 업데이트하고,

상기 파라미터 연산부는

상기 적어도 하나의 워커 프로세스로부터 상기 적어도 하나의 워커 영역에 상기 적어도 하나의 워커 그래디언트 파라미터가 업데이트되었음을 알림받으면, 상기 적어도 하나의 워커 그래디언트 파라미터를 상기 마스터 그래디언트 파라미터에 누적 연산하여 업데이트하고,

상기 마스터 프로세스는

업데이트된 상기 마스터 그래디언트 파라미터를 이용하여 상기 마스터 가중치 파라미터를 업데이트하는 것을 특징으로 하는 파라미터 서버.

【직권보정 3】

【보정항목】청구범위

【보정세부항목】청구항 18

【변경전】

제8항에 있어서,

상기 마스터 프로세스 및 워커 프로세스는,

상기 원격 직접 메모리 접근(RDMA)을 지원하는 고속 네트워크를 통하여, 상기 파라미터 서버에 저장한 상기 분산 딥러닝 파라미터를 직접 읽어오거나 쓰는 방식으로 상기 분산 딥러닝 파라미터를 공유하는 것을 특징으로 하는 파라미터 서버.

【변경후】

제8항에 있어서,

상기 마스터 프로세스 및 워커 프로세스는,

원격 직접 메모리 접근(RDMA)을 지원하는 고속 네트워크를 통하여, 상기 파라미터 서버에 저장한 상기 분산 딥러닝 파라미터를 직접 읽어오거나 쓰는 방식으로 상기 분산 딥러닝 파라미터를 공유하는 것을 특징으로 하는 파라미터 서버.