



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0017048
(43) 공개일자 2023년02월03일

(51) 국제특허분류(Int. Cl.)
G06F 17/16 (2006.01) G06F 7/78 (2022.01)
G06F 9/48 (2018.01)
(52) CPC특허분류
G06F 17/16 (2013.01)
G06F 7/78 (2022.01)
(21) 출원번호 10-2021-0098783
(22) 출원일자 2021년07월27일
심사청구일자 2021년07월27일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
김지훈
서울특별시 서초구 서초대로38길 12, 102동 603호 (서초동, 마제스타시티, 힐스테이트 서리풀)
김현지
경기도 성남시 분당구 야탑로 20, 111동 903호(야탑동, 탑마을선경아파트)
(74) 대리인
리앤목특허법인

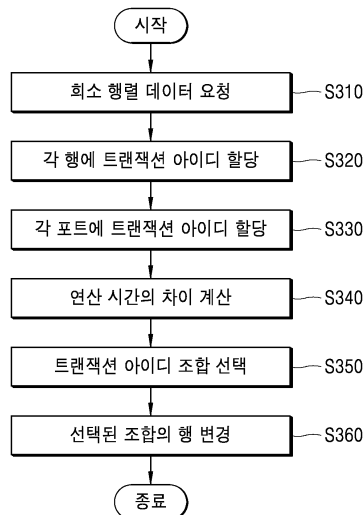
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 연산 가속 방법, 연산 가속 장치 및 상기 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램

(57) 요약

최소 행렬의 연산을 가속하는 방법, 연산 가속 장치 및 상기 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램을 위하여, 본 발명에 따른 컴퓨팅 장치에 의해 수행되는 최소 행렬의 연산을 가속하는 방법은, 상기 최소 행렬의 데이터를 메모리에 요청하는 단계, 상기 최소 행렬의 각 행에 상기 각 행에 대응되는 트랜잭션 아이디를 할당하는 단계, 복수의 메모리 액세스 포트에 상기 트랜잭션 아이디를 맵핑하여 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하는 단계, 상기 각 메모리 액세스 포트에 대하여 상기 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하고, 상기 각 메모리 액세스 포트 사이의 상기 연산 시간의 차이를 계산하는 단계, 및 상기 연산 시간의 차이에 기초하여 각 트랜잭션 아이디 사이의 조합들 중에 하나의 조합을 선택하고, 선택된 조합의 상기 트랜잭션 아이디에 할당된 각 행을 서로 변경하는 단계를 포함한다.

대표도 - 도3



(52) CPC특허분류

G06F 9/4881 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711120090
과제번호	2020-0-01847-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	비대면 · 인공지능 사회를 위한 반도체 시스템 융합혁신기술 개발
기 여 율	1/1
과제수행기관명	한국과학기술원
연구기간	2020.07.01 ~ 2025.12.31

명세서

청구범위

청구항 1

컴퓨팅 장치에 의해 수행되는 회소 행렬의 연산을 가속하는 방법에 있어서,

상기 회소 행렬의 데이터를 메모리에 요청하는 단계;

상기 회소 행렬의 각 행에 상기 각 행에 대응되는 트랜잭션 아이디를 할당하는 단계;

복수의 메모리 액세스 포트에 상기 트랜잭션 아이디를 맵핑하여 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하는 단계;

상기 각 메모리 액세스 포트에 대하여 상기 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하고, 상기 각 메모리 액세스 포트 사이의 상기 연산 시간의 차이를 계산하는 단계; 및

상기 연산 시간의 차이에 기초하여 각 트랜잭션 아이디 사이의 조합들 중에 하나의 조합을 선택하고, 선택된 조합의 상기 트랜잭션 아이디에 할당된 각 행을 서로 변경하는 단계를 포함하는 연산 가속 방법.

청구항 2

제1항에 있어서,

상기 각 행에 대응되는 트랜잭션 아이디를 할당하는 단계는,

상기 회소 행렬의 각 행에 대하여 행 순서대로 행 번호를 설정하는 단계; 및

상기 행 번호에 각각 대응되는 서로 다른 트랜잭션 아이디를 각 행마다 할당하는 단계를 포함하는 연산 가속 방법.

청구항 3

제1항에 있어서,

상기 복수의 메모리 액세스 포트는 서로 다른 제1 포트 및 제2 포트를 포함하고,

상기 트랜잭션 아이디는 서로 다른 제1 아이디, 제2 아이디, 제3 아이디 및 제4 아이디를 포함하는 연산 가속 방법.

청구항 4

제3항에 있어서,

상기 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하는 단계는,

상기 제1 포트에 상기 제1 아이디 및 상기 제2 아이디를 맵핑하고, 상기 제2 포트에 상기 제3 아이디 및 제4 아이디를 맵핑하는 단계; 및

상기 제1 포트 및 상기 제2 포트에 상기 제1 아이디, 상기 제2 아이디, 상기 제3 아이디 및 상기 제4 아이디를 할당하는 단계를 포함하는 연산 가속 방법.

청구항 5

제1항에 있어서,

상기 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하는 단계는, 상기 각 메모리 액세스 포트에 동일한 개수의 상기 트랜잭션 아이디를 각각 할당하는 단계인 연산 가속 방법.

청구항 6

제1항에 있어서,

상기 연산 시간의 차이를 계산하는 단계는,

상기 각 메모리 액세스 포트에 할당되는 상기 트랜잭션 아이디 각각의 연산 시간을 합하여 상기 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하는 단계; 및

상기 각 메모리 액세스 포트에 할당되는 연산 시간 사이의 차를 계산하는 단계를 포함하는 연산 가속 방법.

청구항 7

제1항에 있어서,

상기 트랜잭션 아이디에 할당된 각 행을 서로 변경하는 단계는,

복수의 트랜잭션 아이디 중 서로 다른 2개의 트랜잭션 아이디의 조합으로 이루어지는 스케줄링 조합을 생성하는 단계;

상기 연산 시간의 차이를 절반으로 나눈 값을 이동 사이즈로 설정하는 단계; 및

상기 스케줄링 조합 중에 상기 서로 다른 2개의 트랜잭션 아이디의 연산 시간 사이의 차가 상기 이동 사이즈와 가장 가까운 상기 스케줄링 조합을 선택하는 단계를 포함하는 연산 가속 방법.

청구항 8

컴퓨터를 이용하여 제1항 내지 제7항 중 어느 한 항의 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램.

청구항 9

적어도 하나의 프로세서를 포함하고,

상기 프로세서는, 최소 행렬의 데이터를 메모리에 요청하고, 상기 최소 행렬의 각 행에 상기 각 행에 대응되는 트랜잭션 아이디를 할당하고, 복수의 메모리 액세스 포트에 상기 트랜잭션 아이디를 맵핑하여 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하고, 상기 각 메모리 액세스 포트에 대하여 상기 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하고, 상기 각 메모리 액세스 포트 사이의 상기 연산 시간의 차이를 계산하고, 상기 연산 시간의 차이에 기초하여 각 트랜잭션 아이디 사이의 조합들 중에 하나의 조합을 선택하고, 선택된 조합의 상기 트랜잭션 아이디에 할당된 각 행을 서로 변경하는 연산 가속 장치.

청구항 10

제9항에 있어서,

상기 트랜잭션 아이디는 상기 최소 행렬의 행 번호에 각각 대응되는 서로 다른 트랜잭션 아이디가 각 행마다 할당되고,

상기 행 번호는 상기 최소 행렬의 각 행에 대하여 행 순서대로 설정되는 연산 가속 장치.

청구항 11

제9항에 있어서,

상기 복수의 메모리 액세스 포트는 서로 다른 제1 포트 및 제2 포트를 포함하고,

상기 트랜잭션 아이디는 서로 다른 제1 아이디, 제2 아이디, 제3 아이디 및 제4 아이디를 포함하는 연산 가속 장치.

청구항 12

제11항에 있어서,

상기 트랜잭션 아이디는 상기 제1 포트 및 상기 제2 포트에 상기 제1 아이디, 상기 제2 아이디, 상기 제3 아이디 및 상기 제4 아이디가 할당되고,

상기 제1 포트에 상기 제1 아이디 및 상기 제2 아이디가 맵핑되고, 상기 제2 포트에 상기 제3 아이디 및 제4 아이디가 맵핑되는 연산 가속 장치.

청구항 13

제9항에 있어서,

상기 트랜잭션 아이디는 상기 각 메모리 액세스 포트에 동일한 개수의 상기 트랜잭션 아이디가 각각 할당되는 연산 가속 장치.

청구항 14

제9항에 있어서,

상기 연산 시간의 차이는 상기 각 메모리 액세스 포트에 할당되는 연산 시간 사이의 차로 계산되고,

상기 각 메모리 액세스 포트에 할당되는 연산 시간은 상기 각 메모리 액세스 포트에 할당되는 상기 트랜잭션 아이디 각각의 연산 시간을 합하여 계산되는 연산 가속 장치.

청구항 15

제9항에 있어서,

상기 트랜잭션 아이디에 할당된 각 행의 변경은, 복수의 트랜잭션 아이디 중 서로 다른 2개의 트랜잭션 아이디의 조합으로 이루어지는 스케줄링 조합 중에 상기 서로 다른 2개의 트랜잭션 아이디의 연산 시간 사이의 차이가 이동 사이즈와 가장 가까운 상기 스케줄링 조합의 상기 트랜잭션 아이디에 할당된 각 행이 서로 변경되고,

상기 이동 사이즈는 상기 연산 시간의 차이를 절반으로 나눈 값으로 설정되는 연산 가속 장치.

발명의 설명

기술 분야

[0001] 본 발명의 실시예들은, 최소 행렬-벡터 곱셈의 연산에 있어서 연산 가속 방법, 연산 가속 장치 및 상기 방법을

실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램에 관한 것이다.

배경 기술

- [0002] 최근 많은 국가에서 과학, 산업, 국가 안보와 같은 다양한 분야를 강화하기 위해 고성능 컴퓨팅에 많은 관심을 가지고 있다. 고성능 컴퓨팅에는 기존의 과학 컴퓨팅뿐만 아니라 인공지능 학습 및 데이터 분석 방법의 응용 프로그램도 포함된다.
- [0003] 희소 행렬(Sparse matrix)은 행렬의 값이 대부분 0인 행렬을 의미한다. 희소 행렬 연산은 기존의 조밀 행렬 연산과 다르게 메모리 접근량 대비 연산량이 적어 기존의 고성능 컴퓨팅 시스템에서 성능이 저하되고 있어 최근 고성능 컴퓨팅과 인공지능 어플리케이션에서 희소 행렬 연산의 중요성이 대두되고 있다.
- [0004] 희소 행렬 연산 중에서도 희소 행렬-벡터 곱셈(Sparse matrix-vector multiplication)은 기존의 고성능 과학 연산부터 최근 주목 받고있는 머신 러닝과 딥 러닝에서도 중요한 연산 중 하나이다.
- [0005] 다만, 희소 행렬-벡터 곱셈은 간접 메모리 접근과 불규칙한 메모리 접근성을 가지고 있어 기존의 메모리 시스템에서 메모리 접근 수가 증가하거나 메모리 접근 시간이 증가하여 전체 연산 성능이 저하된다.

발명의 내용

해결하려는 과제

- [0006] 본 발명이 해결하고자 하는 과제는 전술한 문제점을 극복하기 위한 것으로서, 희소 행렬-벡터 곱셈 연산에 있어서 연산 성능을 향상시킬 수 있는 연산 가속 방법, 컴퓨터 프로그램 및 장치를 제공하는 것이다. 그러나 이러한 과제는 예시적인 것으로, 이에 의해 본 발명의 범위가 한정되는 것은 아니다.

과제의 해결 수단

- [0007] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 본 발명의 일 측면에 따른 컴퓨팅 장치에 의해 수행되는 희소 행렬의 연산을 가속하는 방법은, 상기 희소 행렬의 데이터를 메모리에 요청하는 단계, 상기 희소 행렬의 각 행에 상기 각 행에 대응되는 트랜잭션 아이디를 할당하는 단계, 복수의 메모리 액세스 포트에 상기 트랜잭션 아이디를 맵핑하여 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하는 단계, 상기 각 메모리 액세스 포트에 대하여 상기 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하고, 상기 각 메모리 액세스 포트 사이의 상기 연산 시간의 차이를 계산하는 단계, 및 상기 연산 시간의 차이에 기초하여 각 트랜잭션 아이디 사이의 조합들 중에 하나의 조합을 선택하고, 선택된 조합의 상기 트랜잭션 아이디에 할당된 각 행을 서로 변경하는 단계를 포함한다.
- [0008] 일 예에 따르면, 상기 각 행에 대응되는 트랜잭션 아이디를 할당하는 단계는, 상기 희소 행렬의 각 행에 대하여 행 순서대로 행 번호를 설정하는 단계, 및 상기 행 번호에 각각 대응되는 서로 다른 트랜잭션 아이디를 각 행마다 할당하는 단계를 포함할 수 있다.
- [0009] 다른 예에 따르면, 상기 복수의 메모리 액세스 포트는 서로 다른 제1 포트 및 제2 포트를 포함하고, 상기 트랜잭션 아이디는 서로 다른 제1 아이디, 제2 아이디, 제3 아이디 및 제4 아이디를 포함할 수 있다.
- [0010] 또 다른 예에 따르면, 상기 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하는 단계는, 상기 제1 포트에 상기 제1 아이디 및 상기 제2 아이디를 맵핑하고, 상기 제2 포트에 상기 제3 아이디 및 제4 아이디를 맵핑하는 단계, 및 상기 제1 포트 및 상기 제2 포트에 상기 제1 아이디, 상기 제2 아이디, 상기 제3 아이디 및 상기 제4 아이디를 할당하는 단계를 포함할 수 있다.
- [0011] 또 다른 예에 따르면, 상기 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하는 단계는, 상기 각 메모리 액세스 포트에 동일한 개수의 상기 트랜잭션 아이디를 각각 할당하는 단계일 수 있다.
- [0012] 또 다른 예에 따르면, 상기 연산 시간의 차이를 계산하는 단계는, 상기 각 메모리 액세스 포트에 할당되는 상기 트랜잭션 아이디 각각의 연산 시간을 합하여 상기 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하는 단계, 및 상기 각 메모리 액세스 포트에 할당되는 연산 시간 사이의 차를 계산하는 단계를 포함할 수 있다.
- [0013] 또 다른 예에 따르면, 상기 트랜잭션 아이디에 할당된 각 행을 서로 변경하는 단계는, 복수의 트랜잭션 아이디 중 서로 다른 2개의 트랜잭션 아이디의 조합으로 이루어지는 스케줄링 조합을 생성하는 단계, 상기 연산 시간의 차이를 절반으로 나눈 값을 이동 사이즈로 설정하는 단계, 및 상기 스케줄링 조합 중에 상기 서로 다른 2개의

트랜잭션 아이디의 연산 시간 사이의 차가 상기 이동 사이즈와 가장 가까운 상기 스케줄링 조합을 선택하는 단계를 포함할 수 있다.

- [0014] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 컴퓨팅 장치를 이용하여 상술한 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램이 제공된다.
- [0015] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 본 발명의 일 측면에 따른 연산 가속 장치는 적어도 하나의 프로세서를 포함하고, 상기 프로세서는, 최소 행렬의 데이터를 메모리에 요청하고, 상기 최소 행렬의 각 행에 상기 각 행에 대응되는 트랜잭션 아이디를 할당하고, 복수의 메모리 액세스 포트에 상기 트랜잭션 아이디를 맵핑하여 각 메모리 액세스 포트에 상기 트랜잭션 아이디를 할당하고, 상기 각 메모리 액세스 포트에 대하여 상기 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하고, 상기 각 메모리 액세스 포트 사이의 상기 연산 시간의 차이를 계산하고, 상기 연산 시간의 차이에 기초하여 각 트랜잭션 아이디 사이의 조합들 중에 하나의 조합을 선택하고, 선택된 조합의 상기 트랜잭션 아이디에 할당된 각 행을 서로 변경한다.
- [0016] 일 예에 따르면, 상기 트랜잭션 아이디는 상기 최소 행렬의 행 번호에 각각 대응되는 서로 다른 트랜잭션 아이디가 각 행마다 할당되고, 상기 행 번호는 상기 최소 행렬의 각 행에 대하여 행 순서대로 설정될 수 있다.
- [0017] 다른 예에 따르면, 상기 복수의 메모리 액세스 포트는 서로 다른 제1 포트 및 제2 포트를 포함하고, 상기 트랜잭션 아이디는 서로 다른 제1 아이디, 제2 아이디, 제3 아이디 및 제4 아이디를 포함할 수 있다.
- [0018] 또 다른 예에 따르면, 상기 트랜잭션 아이디는 상기 제1 포트 및 상기 제2 포트에 상기 제1 아이디, 상기 제2 아이디, 상기 제3 아이디 및 상기 제4 아이디가 할당되고, 상기 제1 포트에 상기 제1 아이디 및 상기 제2 아이디가 맵핑되고, 상기 제2 포트에 상기 제3 아이디 및 제4 아이디가 맵핑될 수 있다.
- [0019] 또 다른 예에 따르면, 상기 트랜잭션 아이디는 상기 각 메모리 액세스 포트에 동일한 개수의 상기 트랜잭션 아이디가 각각 할당될 수 있다.
- [0020] 또 다른 예에 따르면, 상기 연산 시간의 차이는 상기 각 메모리 액세스 포트에 할당되는 연산 시간 사이의 차로 계산되고, 상기 각 메모리 액세스 포트에 할당되는 연산 시간은 상기 각 메모리 액세스 포트에 할당되는 상기 트랜잭션 아이디 각각의 연산 시간을 합하여 계산될 수 있다.
- [0021] 또 다른 예에 따르면, 상기 트랜잭션 아이디에 할당된 각 행의 변경은, 복수의 트랜잭션 아이디 중 서로 다른 2개의 트랜잭션 아이디의 조합으로 이루어지는 스케줄링 조합 중에 상기 서로 다른 2개의 트랜잭션 아이디의 연산 시간 사이의 차가 이동 사이즈와 가장 가까운 상기 스케줄링 조합의 상기 트랜잭션 아이디에 할당된 각 행이 서로 변경되고, 상기 이동 사이즈는 상기 연산 시간의 차이를 절반으로 나눈 값으로 설정될 수 있다.
- [0022] 전술한 것 외의 다른 측면, 특징, 이점은 이하의 발명을 실시하기 위한 구체적인 내용, 청구범위 및 도면으로부터 명확해질 것이다.

발명의 효과

- [0023] 상기한 바와 같이 이루어진 본 발명의 일 실시예에 따르면, 최소 행렬-벡터 곱셈 연산의 연산 성능을 향상시킬 수 있는 연산 가속 장치, 연산 가속 방법 및 상기 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램을 구현할 수 있다. 물론 이러한 효과에 의해 본 발명의 범위가 한정되는 것은 아니다.

도면의 간단한 설명

- [0024] 도 1은 본 발명의 일 실시예에 따른 연산 가속 장치가 메모리에 액세스하는 회로를 개략적으로 도시하는 도면이다.
- 도 2는 도 1의 프로세서의 구성을 설명하기 위한 도면이다.
- 도 3은 본 발명의 일 실시예에 따른 연산 가속 방법을 설명하기 위한 순서도이다.
- 도 4는 본 발명의 일 실시예에 따른 트랜잭션 아이디가 할당되는 방법을 설명하기 위한 도면이다.
- 도 5는 본 발명의 일 실시예에 따른 각 메모리 액세스 포트 사이의 연산 시간의 차이를 계산하는 방법을 설명하기 위한 도면이다.
- 도 6은 본 발명의 일 실시예에 따른 트랜잭션 아이디 조합을 선택하는 방법을 설명하기 위한 도면이다.

도 7은 본 발명의 일 실시예에 따라 서로 변경된 행을 설명하기 위한 도면이다.

도 8 내지 도 10은 본 발명의 일 실시예에 따른 연산 가속 장치의 성능을 실험한 실험 데이터 및 결과의 예시들이다.

발명을 실시하기 위한 구체적인 내용

- [0025] 아래에서는 첨부한 도면을 참조하여 본 개시가 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 다양한 실시예들을 상세히 설명한다. 그러나 본 개시의 기술적 사상은 다양한 형태로 변형되어 구현될 수 있으므로 본 명세서에서 설명하는 실시예들로 제한되지 않는다. 본 명세서에 개시된 실시예들을 설명함에 있어서 관련된 공지 기술을 구체적으로 설명하는 것이 본 개시의 기술적 사상의 요지를 흐릴 수 있다고 판단되는 경우 그 공지 기술에 대한 구체적인 설명을 생략한다. 동일하거나 유사한 구성요소는 동일한 참조 번호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다.
- [0026] 본 명세서에서 어떤 요소가 다른 요소와 "연결"되어 있다고 기술될 때, 이는 "직접적으로 연결"되어 있는 경우 뿐 아니라 그 중간에 다른 요소를 사이에 두고 "간접적으로 연결"되어 있는 경우도 포함한다. 어떤 요소가 다른 요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 요소 외에 또 다른 요소를 배제하는 것이 아니라 또 다른 요소를 더 포함할 수 있는 것을 의미한다.
- [0027] 일부 실시예들은 기능적인 블록 구성들 및 다양한 처리 단계들로 설명될 수 있다. 이러한 기능 블록들의 일부 또는 전부는 특정 기능을 실행하는 다양한 개수의 하드웨어 및/또는 소프트웨어 구성들로 구현될 수 있다. 예를 들어, 본 개시의 기능 블록들은 하나 이상의 마이크로프로세서들에 의해 구현되거나, 소정의 기능을 위한 회로 구성들에 의해 구현될 수 있다. 본 개시의 기능 블록들은 다양한 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 본 개시의 기능 블록들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다. 본 개시의 기능 블록이 수행하는 기능은 복수의 기능 블록에 의해 수행되거나, 본 개시에서 복수의 기능 블록이 수행하는 기능들은 하나의 기능 블록에 의해 수행될 수도 있다. 또한, 본 개시는 전자적인 환경 설정, 신호 처리, 및/또는 데이터 처리 등을 위하여 종래 기술을 채용할 수 있다.
- [0028] 도 1은 본 발명의 일 실시예에 따른 연산 가속 장치가 메모리에 액세스하는 회로를 개략적으로 도시하는 도면이고, 도 2는 도 1의 프로세서의 구성을 설명하기 위한 도면이다.
- [0029] 먼저, 도 1을 참조하면, 본 발명의 일 실시예에 따른 연산 가속 장치(100)는 메모리(50)에 액세스할 수 있다. 예를 들어, 연산 가속 장치(100)는 메모리(50)에 액세스하여 데이터의 읽기나 쓰기를 할 수 있다. 예컨대, 메모리(50)는 프로세서(110)가 처리하는 데이터를 일시적 또는 영구적으로 저장할 수 있다. 메모리(50)는 RAM(Random Access Memory), ROM(Read Only Memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(Permanent Mass Storage Device)를 포함할 수 있으나, 본 발명이 이에 한정되는 것은 아니다.
- [0030] 도 1에 도시된 바와 같이, 연산 가속 장치(100)는 프로세서(110), 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)를 포함한다. 하지만 본 발명은 이에 한정되지 않으며, 연산 가속 장치(100)는 다른 구성요소를 더 포함할 수도 있고 일부 구성요소가 생략될 수도 있다. 연산 가속 장치(100)의 일부 구성요소는 복수의 장치로 분리될 수도 있고, 복수개의 구성요소들이 하나의 장치로 병합될 수도 있다.
- [0031] 프로세서(110)는 메모리(50)에 저장된 데이터(예컨대, 회소 행렬)의 검색, 추출, 연산, 읽기, 쓰기 등의 동작을 수행할 수 있다. 프로세서(110)는 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 포함할 수 있으나, 본 발명은 이에 한정되지 않는다.
- [0032] 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)는 연산 가속 장치(100)와 메모리(50) 사이를 서로 전기적으로 연결하여 연산 가속 장치(100)와 메모리(50) 사이에 데이터를 주고받을 수 있는 통로이다. 도 1에는 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)가 도시되어 있지만, 본 발명은 이에 한정되지 않으며 연산 가속 장치(100)와 메모리(50) 사이에 복수의 메모리 액세스 포트가 포함될 수 있다.
- [0033] 도 1에 도시된 바와 같이, 프로세서(110)는 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)와 서로 전기적으로 연결되어 데이터를 주고 받을 수 있다. 구체적으로, 프로세서(110)는 제1 메모리 액세스 포트(130)와 제1 읽기 회로(r1)로 연결되어 제1 메모리 액세스 포트(130)를 통해 메모리(50)에 저장된 데이터를 읽을 수 있다. 또한, 프로세서(110)는 제2 메모리 액세스 포트(150)와 제2 읽기 회로(r2)로 연결되어 제2 메모리

엑세스 포트(150)를 통해 메모리(50)에 저장된 데이터를 읽을 수 있다.

- [0034] 또한, 도 1에 도시된 바와 같이, 프로세서(110)는 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)와 제1 쓰기 회로(w1)로 연결되어 제1 쓰기 회로(w1)를 통해 메모리(50)에 데이터를 쓸 수 있다. 또한, 프로세서(110)는 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)와 제2 쓰기 회로(w2)로 연결되어 제2 쓰기 회로(w2)를 통해 메모리(50)에 데이터를 쓸 수 있다.
- [0035] 도 1에 도시된 것과 같은 연산 가속 장치(100)는 프로세서(110)의 처리 또는 제어를 위한 프로그램 등, 전반의 동작을 위한 다양한 데이터를 저장한 저장 매체(미도시)를 더 포함할 수 있다. 저장 매체는 프로세서(110)에서 구동되는 동작을 위한 데이터들, 그리고 명령어들을 저장할 수 있다.
- [0036] 도 2를 참조하면, 프로세서(110)는 데이터 요청 모듈(111), 트랜잭션 아이디 할당 모듈(113), 계산 모듈(115) 및 행 변경 모듈(117)을 포함한다.
- [0037] 프로세서(110)와 이에 포함된 모듈들은, 전적으로 하드웨어일 수 있고, 또는 부분적으로 하드웨어이고 부분적으로 소프트웨어인 측면을 가질 수 있다. 예컨대, 프로세서(110)와 이에 포함된 각 모듈은 특정 형식 및 내용의 데이터를 처리하거나 전자통신 방식으로 주고받기 위한 하드웨어 및 이에 관련된 소프트웨어를 통칭할 수 있다. 본 명세서에서 "모듈" 또는 "장치" 등의 용어는 하드웨어 및 해당 하드웨어에 의해 구동되는 소프트웨어의 조합을 지칭하는 것으로 해석될 수 있다. 예컨대, 하드웨어는 CPU 또는 다른 프로세서(processor)를 포함하는 데이터 처리 기기일 수 있다. 또한, 하드웨어에 의해 구동되는 소프트웨어는 실행중인 프로세스, 객체(object), 실행파일(executable file), 실행 스레드(thread of execution), 프로그램(program) 등을 지칭할 수 있다.
- [0038] 또한, 프로세서(110)를 구성하는 각각의 요소는 반드시 서로 물리적으로 구분되는 별개의 장치를 지칭하는 것으로 한정되는 것은 아니다. 즉, 도 2의 데이터 요청 모듈(111), 트랜잭션 아이디 할당 모듈(113), 계산 모듈(115) 및 행 변경 모듈(117) 등은 프로세서(110)에 의해 수행되는 동작에 따라 기능적으로 구분한 것일 뿐, 반드시 서로 독립적으로 구분되어야 하는 것이 아니다. 물론, 실시예에 따라서는 이들은 서로 물리적으로 구분되는 별개의 장치로 구현되는 것도 가능하다.
- [0039] 이하에서는 상술한 것과 같은 구성을 취할 수 있는 프로세서(110)의 각 모듈들의 동작 등에 대해, 연산 가속 방법과 함께 설명한다.
- [0040] 도 3은 본 발명의 일 실시예에 따른 연산 가속 방법을 설명하기 위한 순서도이다. 또한, 도 4 내지 도 7은 본 발명의 일 실시예에 따른 연산 가속 방법을 설명하기 위한 도면들이다.
- [0041] 도 3에 도시된 것과 같이, 본 실시예에 따른 연산 가속 방법은 메모리에 저장된 회소 행렬의 데이터를 메모리에 요청하는 단계(S310)를 거친다.
- [0042] 도 1 및 도 2를 함께 참조하면, 데이터 요청 모듈(111)은 메모리(50)에 액세스하여 메모리(50)에 저장된 회소 행렬의 데이터를 요청할 수 있다. 예컨대, 회소 행렬의 데이터는 복수의 행 및 복수의 열을 갖는 데이터를 포함할 수 있다. 또한, 데이터 요청 모듈(111)은 메모리(50)로부터 회소 행렬의 데이터를 획득할 수 있다.
- [0043] 또한, 본 발명의 일 실시예에 따른 연산 가속 방법은 회소 행렬의 각 행에, 각 행에 대응되는 트랜잭션 아이디를 할당하는 단계(S320) 및 복수의 메모리 액세스 포트에 트랜잭션 아이디를 맵핑하여 각 메모리 액세스 포트에 트랜잭션 아이디를 할당하는 단계(S330)를 거친다.
- [0044] 도 4를 함께 참조하면, 도 4는 본 발명의 일 실시예에 따른 트랜잭션 아이디가 할당되는 방법을 설명하기 위한 도면이다.
- [0045] 도 4에 도시된 것과 같이, 회소 행렬(m)은 제1행(row1), 제2행(row2), 제3행(row3) 및 제4행(row4)의 4개의 행을 갖는 행렬일 수 있다.
- [0046] 연산 가속 장치는 복수의 메모리 액세스 포트를 포함할 수 있다. 본 발명의 일 실시예에서, 복수의 메모리 액세스 포트는 물리적으로 서로 다른 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)를 포함할 수 있다. 본 발명의 다른 실시예에서, 복수의 메모리 액세스 포트는 전기적으로 서로 다른 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)를 포함할 수 있다. 본 발명의 또 다른 실시예에서, 복수의 메모리 액세스 포트는 데이터 경로 상으로 서로 다른 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)를 포함할 수 있다.
- [0047] 트랜잭션 아이디 할당 모듈(113)은 트랜잭션 아이디를 할당할 수 있다. 예를 들어, 트랜잭션 아이디는 서로 다

은 제1 트랜잭션 아이디(T1), 제2 트랜잭션 아이디(T2), 제3 트랜잭션 아이디(T3) 및 제4 트랜잭션 아이디(T4)를 포함할 수 있다. 예컨대, 서로 다른 제1 트랜잭션 아이디(T1), 제2 트랜잭션 아이디(T2), 제3 트랜잭션 아이디(T3) 및 제4 트랜잭션 아이디(T4) 각각은 서로 다른 자연수로 표현될 수 있다.

[0048] 트랜잭션 아이디 할당 모듈(113)은 최소 행렬의 각 행에, 각 행에 대응되는 트랜잭션 아이디를 할당할 수 있다. 예를 들어, 제1행(row1)에 제1행(row1)에 대응되는 제1 트랜잭션 아이디(T1)를 할당하고, 제2행(row2)에 제2행(row2)에 대응되는 제2 트랜잭션 아이디(T2)를 할당하고, 제3행(row3)에 제3행(row3)에 대응되는 제3 트랜잭션 아이디(T3)를 할당하고, 제4행(row4)에 제4행(row4)에 대응되는 제4 트랜잭션 아이디(T4)를 할당할 수 있다.

[0049] 또한, 트랜잭션 아이디 할당 모듈(113)은 복수의 메모리 액세스 포트에 트랜잭션 아이디를 맵핑하여 각 메모리 액세스 포트에 트랜잭션 아이디를 할당할 수 있다. 예를 들어, 도 4에 도시된 바와 같이, 트랜잭션 아이디 할당 모듈(113)은 제1 메모리 액세스 포트(130)에 제1 트랜잭션 아이디(T1) 및 제2 트랜잭션 아이디(T2)를 맵핑할 수 있다. 또한, 트랜잭션 아이디 할당 모듈(113)은 제2 메모리 액세스 포트(150)에 제3 트랜잭션 아이디(T3) 및 제4 트랜잭션 아이디(T4)를 맵핑할 수 있다. 이어서, 트랜잭션 아이디 할당 모듈(113)은 제1 메모리 액세스 포트(130)에 제1 트랜잭션 아이디(T1) 및 제2 트랜잭션 아이디(T2)를 할당할 수 있다. 또한, 트랜잭션 아이디 할당 모듈(113)은 제2 메모리 액세스 포트(150)에 제3 트랜잭션 아이디(T3) 및 제4 트랜잭션 아이디(T4)를 할당할 수 있다.

[0050] 본 발명의 일 실시예에서, 각 행에 대응되는 트랜잭션 아이디를 할당하는 단계(S320)는, 최소 행렬의 각 행에 대하여 행 순서대로 행 번호를 설정하는 단계, 및 행 번호에 각각 대응되는 서로 다른 트랜잭션 아이디를 각 행마다 할당하는 단계를 포함할 수 있다. 예를 들어, 도 4에 도시된 바와 같이, 트랜잭션 아이디 할당 모듈(113)은 제1행(row1), 제2행(row2), 제3행(row3) 및 제4행(row4)에 대하여 행 순서대로 1, 2, 3 및 4의 행 번호를 설정하고, 행 번호 1에 대응되는 제1 트랜잭션 아이디(T1)를 제1행(row1)에 할당할 수 있다. 또한, 행 번호 2에 대응되는 제2 트랜잭션 아이디(T2)를 제2행(row2)에 할당할 수 있다. 또한, 행 번호 3에 대응되는 제3 트랜잭션 아이디(T3)를 제3행(row3)에 할당할 수 있다. 또한, 행 번호 4에 대응되는 제4 트랜잭션 아이디(T4)를 제4행(row4)에 할당할 수 있다.

[0051] 본 발명의 일 실시예에서, 각 메모리 액세스 포트에 트랜잭션 아이디를 할당하는 단계(S330)는, 각 메모리 액세스 포트에 동일한 개수의 트랜잭션 아이디를 각각 할당하는 단계일 수 있다. 예를 들어, 도 4에 도시된 바와 같이, 트랜잭션 아이디 할당 모듈(113)은 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150) 각각에 2개씩 동일한 개수의 트랜잭션 아이디를 할당할 수 있다.

[0052] 도 5를 함께 참조하면, 도 5는 본 발명의 일 실시예에 따른 각 메모리 액세스 포트 사이의 연산 시간의 차이를 계산하는 방법을 설명하기 위한 도면이다.

[0053] 도 5 (a)에 도시된 바와 같이, 제1 메모리 액세스 포트(130)에는 제1 트랜잭션 아이디(T1) 및 제2 트랜잭션 아이디(T2)가 할당되고, 제2 메모리 액세스 포트(150)에는 제3 트랜잭션 아이디(T3) 및 제4 트랜잭션 아이디(T4)가 할당될 수 있다.

[0054] 또한, 본 발명의 일 실시예에 따른 연산 가속 방법은 각 메모리 액세스 포트에 대하여 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하고, 각 메모리 액세스 포트 사이의 연산 시간의 차이를 계산하는 단계(S340)를 거친다.

[0055] 계산 모듈(115)은 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150)에 대하여 제1 메모리 액세스 포트(130)에 할당되는 제1 메모리 액세스 포트 연산 시간(TT1)을 계산할 수 있다. 또한, 계산 모듈(115)은 제2 메모리 액세스 포트(150)에 할당되는 제2 메모리 액세스 포트 연산 시간(TT2)을 계산할 수 있다.

[0056] 또한, 계산 모듈(115)은 각 메모리 액세스 포트 사이의 연산 시간의 차이를 계산할 수 있다. 예를 들어, 도 5 (b)에 도시된 바와 같이, 제1 메모리 액세스 포트(130) 및 제2 메모리 액세스 포트(150) 사이의 제1 메모리 액세스 포트 연산 시간(TT1)과 제2 메모리 액세스 포트 연산 시간(TT2) 사이의 차이를 계산할 수 있다.

[0057] 본 발명의 일 실시예에 따른 연산 시간의 차이를 계산하는 단계(S340)는, 각 메모리 액세스 포트에 할당되는 트랜잭션 아이디 각각의 연산 시간을 합하여 각 메모리 액세스 포트에 할당되는 연산 시간을 계산하는 단계, 및 각 메모리 액세스 포트에 할당되는 연산 시간 사이의 차를 계산하는 단계를 포함할 수 있다.

[0058] 예를 들어, 도 5 (a)에 도시된 바와 같이, 제1 메모리 액세스 포트 연산 시간(TT1)은 제1 트랜잭션 아이디(T1)의 연산 시간과 제2 트랜잭션 아이디(T2)의 연산 시간의 합일 수 있다. 여기서, 제1 트랜잭션 아이디(T1)의 연

산 시간은 제1 트랜잭션 아이디(T1)에 할당된 제1행(row1)에 대한 연산 시간을 의미한다. 또한, 제2 트랜잭션 아이디(T2)의 연산 시간은 제2 트랜잭션 아이디(T2)에 할당된 제2행(row2)에 대한 연산 시간을 의미한다.

[0059] 또한, 제2 메모리 액세스 포트 연산 시간(TT2)은 제3 트랜잭션 아이디(T3)의 연산 시간과 제4 트랜잭션 아이디(T4)의 연산 시간의 합일 수 있다. 여기서, 제3 트랜잭션 아이디(T3)의 연산 시간은 제3 트랜잭션 아이디(T3)에 할당된 제3행(row3)에 대한 연산 시간을 의미한다. 또한, 제4 트랜잭션 아이디(T4)의 연산 시간은 제4 트랜잭션 아이디(T4)에 할당된 제4행(row4)에 대한 연산 시간을 의미한다.

[0060] 또한, 계산 모듈(115)은 각 메모리 액세스 포트에 할당되는 연산 시간 사이의 차를 계산할 수 있다. 예를 들어, 도 5 (b)에 도시된 바와 같이, 제1 메모리 액세스 포트 연산 시간(TT1)과 제2 메모리 액세스 포트 연산 시간(TT2) 사이의 차는 연산 시간 차이(d)로 계산될 수 있다.

[0061] 또한, 계산 모듈(115)은 각 메모리 액세스 포트에 할당되는 연산 시간 사이의 차를 절반으로 나눈 값을 계산할 수 있다. 예를 들어, 도 5 (b)에 도시된 바와 같이, 연산 시간 차이(d)의 절반이 이동 사이즈(ms)로 계산될 수 있다.

[0062] 도 6 및 도 7을 함께 참조하면, 도 6은 본 발명의 일 실시예에 따른 트랜잭션 아이디 조합을 선택하는 방법을 설명하기 위한 도면이고, 도 7은 본 발명의 일 실시예에 따라 서로 변경된 행을 설명하기 위한 도면이다.

[0063] 도 6에 도시된 바와 같이, 제1 메모리 액세스 포트(130)에 제1 트랜잭션 아이디(T1) 및 제2 트랜잭션 아이디(T2)가 할당되고, 제2 메모리 액세스 포트(150)에 제3 트랜잭션 아이디(T3) 및 제4 트랜잭션 아이디(T4)가 할당될 수 있다.

[0064] 또한, 본 발명의 일 실시예에 따른 연산 가속 방법은 연산 시간의 차이에 기초하여 각 트랜잭션 아이디 사이의 조합들 중에 하나의 조합을 선택하는 단계(S350) 및 선택된 조합의 트랜잭션 아이디에 할당된 각 행을 서로 변경하는 단계(S360)를 거친다.

[0065] 행 변경 모듈(117)은 연산 시간의 차이에 기초하여 각 트랜잭션 아이디 사이의 조합들 중에 하나의 조합을 선택할 수 있다. 예를 들어, 도 6에 도시된 바와 같이, 도 5 (b)에서 연산 시간 차이(d)에 기초하여 연산 시간 차이(d)의 절반인 이동 사이즈(ms)가 계산되고, 행 변경 모듈(117)은 T1-T3 조합, T1-T4 조합, T2-T3 조합, 및 T2-T4 조합 중에서 하나의 조합을 선택할 수 있다.

[0066] 또한, 행 변경 모듈(117)은 선택된 조합의 트랜잭션 아이디에 할당된 각 행을 서로 변경할 수 있다. 예를 들어, T1-T4 조합이 선택된 경우, T1에 할당된 제1행(row1)과 T4에 할당된 제4행(row4)이 서로 변경될 수 있다.

[0067] 본 발명의 일 실시예에 따른, 트랜잭션 아이디에 할당된 각 행을 서로 변경하는 단계는, 복수의 트랜잭션 아이디 중 서로 다른 2개의 트랜잭션 아이디의 조합으로 이루어지는 스케줄링 조합을 생성하는 단계, 연산 시간의 차이를 절반으로 나눈 값을 이동 사이즈로 설정하는 단계, 및 스케줄링 조합 중에 서로 다른 2개의 트랜잭션 아이디의 연산 시간 사이의 차가 이동 사이즈와 가장 가까운 스케줄링 조합을 선택하는 단계를 포함할 수 있다.

[0068] 행 변경 모듈(117)은 복수의 트랜잭션 아이디 중 서로 다른 2개의 트랜잭션 아이디의 조합으로 이루어지는 스케줄링 조합을 생성할 수 있다. 예를 들어, 도 6에 도시된 바와 같이, T1-T3 조합, T1-T4 조합, T2-T3 조합, 및 T2-T4 조합이 생성될 수 있다.

[0069] 또한, 행 변경 모듈(117)은 연산 시간의 차이를 절반으로 나눈 값을 이동 사이즈로 설정할 수 있다. 예를 들어, 도 6에 도시된 바와 같이, 도 5 (b)에서 연산 시간 차이(d)를 절반으로 나눈 값이 이동 사이즈(ms)로 설정될 수 있다.

[0070] 또한, 행 변경 모듈(117)은 스케줄링 조합 중에 서로 다른 2개의 트랜잭션 아이디의 연산 시간 사이의 차가 이동 사이즈와 가장 가까운 스케줄링 조합을 선택할 수 있다. 예를 들어, 도 6에 도시된 바와 같이, T1-T3 조합, T1-T4 조합, T2-T3 조합, 및 T2-T4 조합에 대하여, T1-T3 연산 시간 사이의 차이, T1-T4 연산 시간 사이의 차이, T2-T3 연산 시간 사이의 차이, 및 T2-T4 연산 시간 사이의 차이가 계산될 수 있다.

[0071] 또한, 도 6에 도시된 바와 같이, 행 변경 모듈(117)은 T1-T4 연산 시간 사이의 차이가 이동 사이즈(ms)와 가장 가까우므로 T1-T4 조합을 선택할 수 있다.

[0072] 본 발명의 다른 실시예에서, 행 변경 모듈(117)은 스케줄링 조합 중에 서로 다른 2개의 트랜잭션 아이디의 연산 시간 사이의 차가 이동 사이즈와 동일한 스케줄링 조합을 선택할 수 있다.

- [0073] 본 발명의 또 다른 실시예에서, 행 변경 모듈(117)은 스케줄링 조합 중에 서로 다른 2개의 트랜잭션 아이디어의 연산 시간 사이의 차가 이동 사이즈와 동일하거나 가장 가까운 스케줄링 조합을 선택할 수 있다.
- [0074] 도 7을 함께 참조하면, 행 변경 모듈(117)은 선택된 조합의 트랜잭션 아이디어에 할당된 각 행을 서로 변경할 수 있다. 예를 들어, 도 7에 도시된 바와 같이, T1-T4 조합이 선택된 경우, T1에 할당된 제1행(row1)과 T4에 할당된 제4행(row4)이 서로 변경될 수 있다. 제1 메모리 액세스 포트 연산 시간(TT1)은 T1과 T4가 변경됨으로써 새로운 제1 메모리 액세스 포트 연산 시간(TT1')으로 계산될 수 있다. 또한, 제2 메모리 액세스 포트 연산 시간(TT2)은 T1과 T4가 변경됨으로써 새로운 제2 메모리 액세스 포트 연산 시간(TT2')으로 계산될 수 있다.
- [0075] 도 8 내지 도 10은 본 발명의 일 실시예에 따른 연산 가속 장치의 성능을 실험한 실험 데이터 및 결과의 예시들이다.
- [0076] 먼저, 도 8은 본 발명의 일 실시예에 따른 연산 가속 장치의 성능을 실험하기 위한 최소 행렬 데이터의 패턴들을 도시한다. 최소 행렬 데이터들은 CSR(Compressed Sparse Row) 형식으로 DRAM에 미리 저장되어 있을 수 있다.
- [0077] 도 9는 도 8의 최소 행렬 데이터들에 대하여 본 발명의 일 실시예에 따른 연산 가속 장치의 트랜잭션 수를 늘렸을 때의 DRAM에 대한 액세스 수를 비교한 그래프를 도시한다.
- [0078] 도 9에 도시된 바와 같이, 메모리 액세스 포트의 갯수를 늘림으로써 연산되는 행의 숫자가 늘어 트랜잭션 수가 증가하면 메모리 액세스 수가 최대 74% 감소한다.
- [0079] 다만, 메모리 액세스 수의 감소율은 트랜잭션 수의 증가에 따라 점점 감소하며, 본 발명에 따른 최적의 트랜잭션 수는 6으로 설정될 수 있다.
- [0080] 도 10은 도 8의 최소 행렬 데이터들에 대하여 본 발명의 일 실시예에 따른 연산 가속 장치의 트랜잭션 수준의 스케줄링 효과를 보여주는 도면이다.
- [0081] 도 10에 도시된 바와 같이, 최소 행렬의 행간에 0이 아닌 요소의 차이가 큰 경우 연산 속도가 최대 14% 향상된다.
- [0082] 본 발명에 따르면, 데이터의 요청과 데이터의 처리를 분리하고, 복수의 메모리 액세스 포트를 통해 여러 행을 한 번에 처리하는 다중 행 처리를 수행함으로써 중복되는 메모리 액세스를 줄여 메모리 액세스 수를 줄일 수 있는 효과가 있다.
- [0083] 또한, 본 발명에 따르면, 연산을 병렬화하는 동안 발생할 수 있는 연산 시간 불균형 문제를 완화하기 위해 트랜잭션 수준의 스케줄링이 적용되었으며, 최소 행렬의 행간에 0이 아닌 요소의 차이가 큰 최소 행렬에 대하여, 더 큰 연산 속도 향상의 효과가 있다.
- [0085] 이상 설명된 다양한 실시예들은 예시적이며, 서로 구별되어 독립적으로 실시되어야 하는 것은 아니다. 본 명세서에서 설명된 실시예들은 서로 조합된 형태로 실시될 수 있다.
- [0086] 이상 설명된 다양한 실시예들은 컴퓨터 상에서 다양한 구성요소를 통하여 실행될 수 있는 컴퓨터 프로그램의 형태로 구현될 수 있으며, 이와 같은 컴퓨터 프로그램은 컴퓨터로 판독 가능한 매체에 기록될 수 있다. 이때, 매체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수개 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 애플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다.
- [0087] 본 명세서에서, "부", "모듈" 등은 프로세서 또는 회로와 같은 하드웨어 구성(hardware component), 및/또는 프로세서와 같은 하드웨어 구성에 의해 실행되는 소프트웨어 구성(software component)일 수 있다. 예를 들면, "부", "모듈" 등은 소프트웨어 구성 요소들, 객체 지향 소프트웨어 구성 요소들, 클래스 구성 요소들 및 태스크 구성 요소들과 같은 구성 요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이

들 및 변수들에 의해 구현될 수 있다.

[0088] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

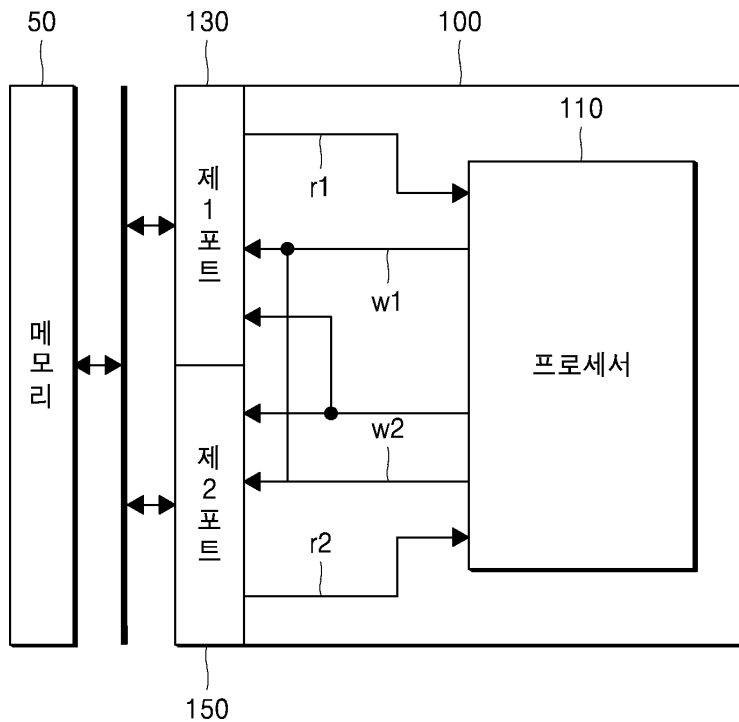
[0089] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

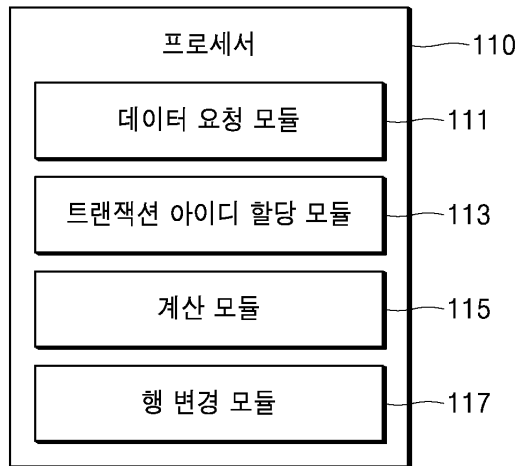
[0090] 50: 메모리
100: 연산 가속 장치
110: 프로세서
130: 제1 메모리 액세스 포트
150: 제2 메모리 액세스 포트

도면

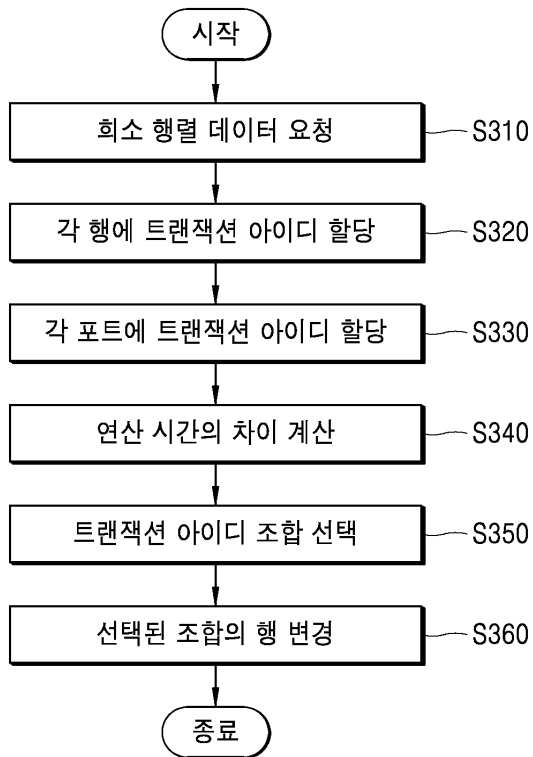
도면1



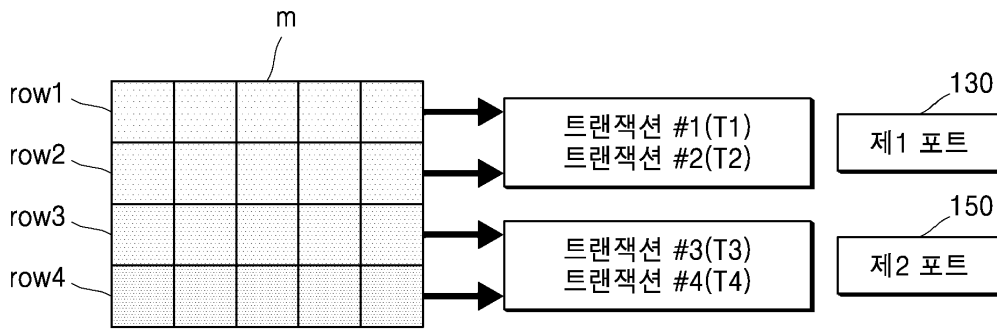
도면2



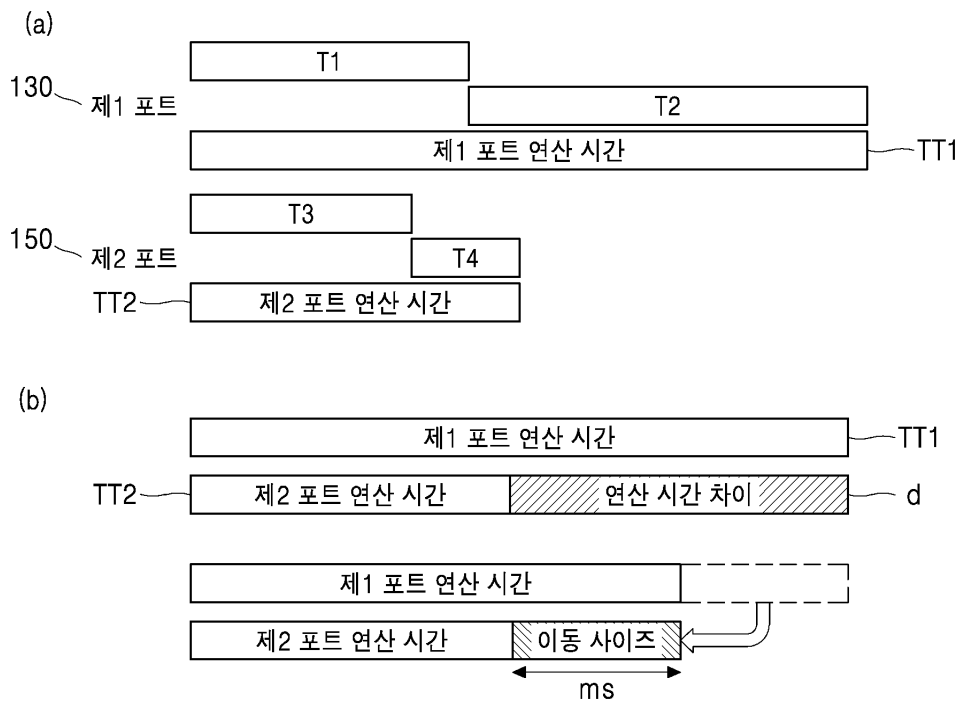
도면3



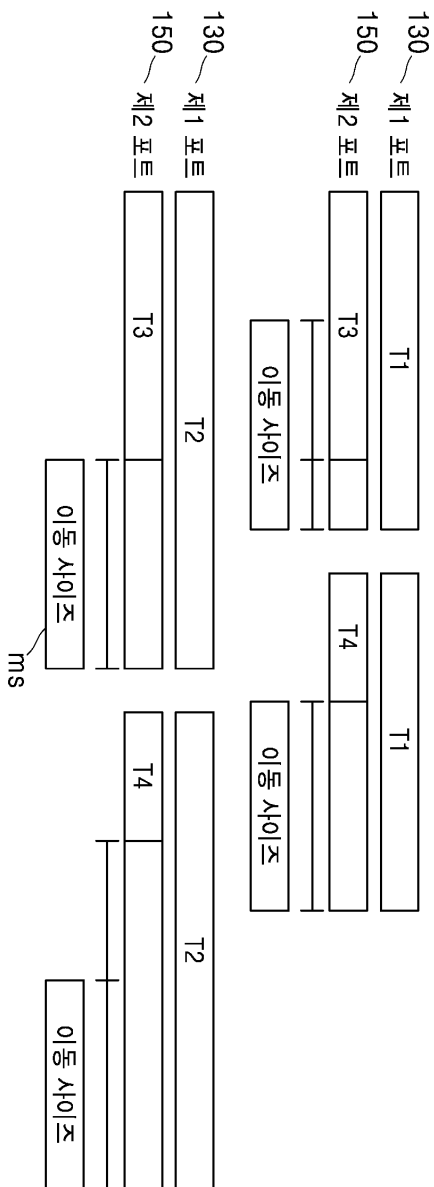
도면4



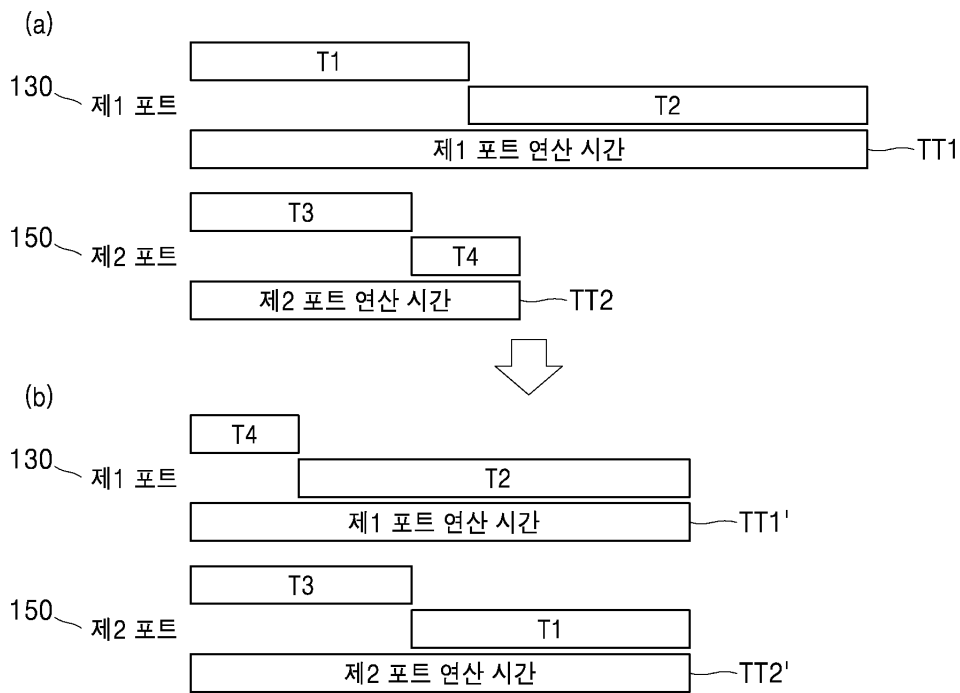
도면5



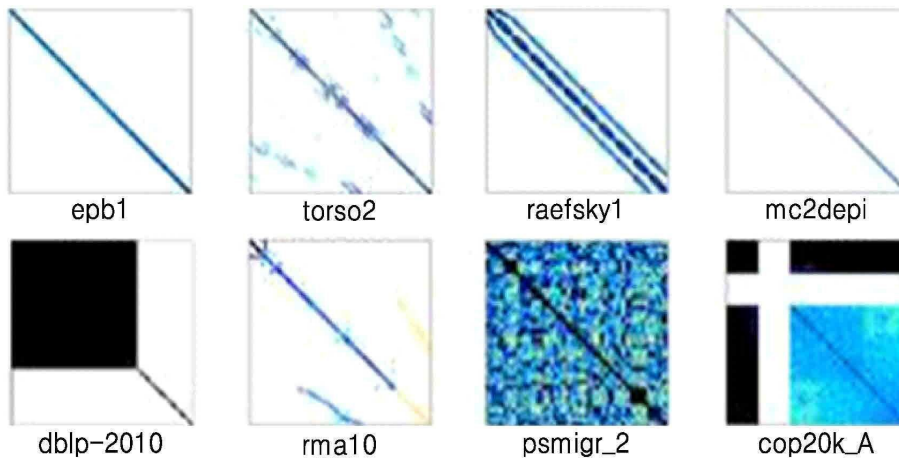
도면6



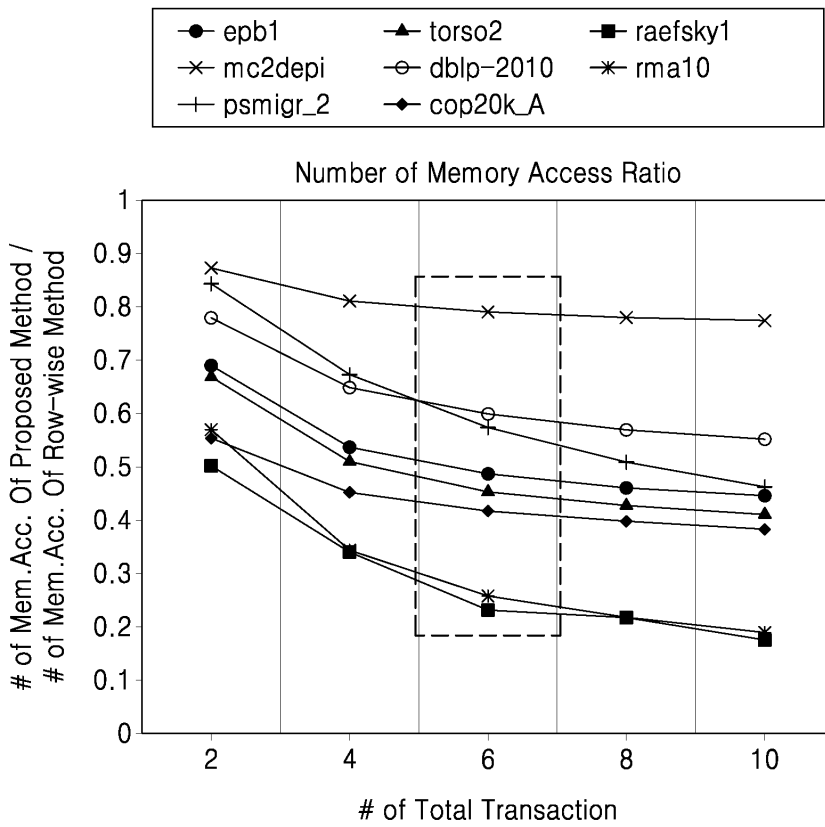
도면7



도면8



도면9



도면10

