(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년06월27일
(11) 등록번호 10-2548234
(24) 등록일자 2023년06월22일

(51) 국제특허분류(Int. Cl.)
*G06N 3/063* (2023.01) *G06N 3/04* (2023.01)
(52) CPC특허분류
*G06N 3/063* (2013.01)
*G06N 3/04* (2023.01)
(21) 출원번호 10-2021-0122414
(22) 출원일자 2021년09월14일
심사청구일자 2021년09월14일
(65) 공개번호 10-2023-0039290
(43) 공개일자 2023년03월21일
(56) 선행기술조사문헌
Y. Hu 등. "FeatGraph: A Flexible and Efficient Backend for Graph Neural Network Systems". arXiv:2008.11359v2*
K. Kiningham 등. "GRIP: A Graph Neural Network Accelerator Architecture". arXiv:2007.13828v2*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
**연세대학교 산학협력단**
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
**이진호**
서울특별시 서대문구 연세로 50, 연세대학교 제4공학관 D702호
**김영석**
서울시 서대문구 연세로 50, 제 4공학관 D703호
*(뒷면에 계속)*
(74) 대리인
**정부연**

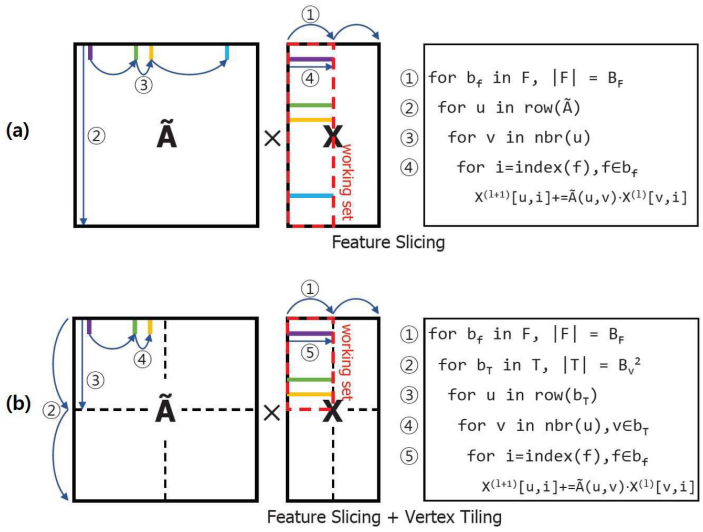
전체 청구항 수 : 총 9 항 심사관 : 노지명

(54) 발명의 명칭 **피처 슬라이싱과 자동 타일 모핑을 이용한 GCN 가속 장치 및 방법**

## (57) 요 약

본 발명은 피처 슬라이싱과 자동 타일 모핑을 이용한 GCN 가속 장치 및 방법에 관한 것으로, 상기 장치는 상기 입력 피처 매트릭스(X)에 관한 피처 슬라이싱을 수행하는 피처 슬라이싱부; 상기 희소 가중 인접 매트릭스(A)에 관한 버텍스 타일링을 수행하는 버텍스 타일링부; 및 상기 피처 슬라이싱에 따른 입력 피처 매트릭스와 상기 버텍스 타일링에 따른 희소 가중 인접 매트릭스 간의 연산을 수행하는 연산부;를 포함한다. 따라서, 본 발명은 피처 슬라이싱과 자동 타일 모핑을 적용하여 GCN 가속의 캐시 동작을 크게 개선하고 더 쉽게 조정할 수 있다.

*대 표 도* - 도6



(52) CPC특허분류
*G06F 2212/302* (2013.01)
(72) 발명자
**유민기**
서울시 서대문구 연세로 50, 제 4공학관 D714호
**송재용**
서울시 서대문구 연세로 50, 제 4공학관 D714호

이 발명을 지원한 국가연구개발사업
과제고유번호 1711126082
과제번호 2020-0-01361-002
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성(R&D)
연구과제명 인공지능대학원지원(연세대학교)
기 여 율 1/2
과제수행기관명 연세대학교 산학협력단
연구기간 2021.01.01 ~ 2021.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711134555
과제번호 2021-0-00853-001
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 신개념PIM반도체선도기술개발(R&D)
연구과제명 PIM 활용을 위한 SW 플랫폼 개발
기 여 율 1/2
과제수행기관명 연세대학교 산학협력단
연구기간 2021.04.01 ~ 2021.12.31
공지예외적용 : 있음

**이정후**
서울시 서대문구 연세로 50, 제 4공학관 D711호

# 명 세 서

## 청구범위

### 청구항 1

희소 가중 인접 매트릭스 및 입력 피처 매트릭스 간의 연산을 수행하는 GCN(Graph Convolutional Networks) 가속 장치에 있어서,

상기 입력 피처 매트릭스에 관한 피처 슬라이싱을 수행하는 피처 슬라이싱부;

상기 희소 가중 인접 매트릭스에 관한 버텍스 타일링을 수행하는 과정에서, 상기 희소 가중 인접 매트릭스를 기초로 거친 모핑(coarse morphing) 단계와 정교 모핑(fine morphing) 단계로 이루어진 자동 타일 모핑을 수행하는 버텍스 타일링부; 및

상기 피처 슬라이싱에 따른 입력 피처 매트릭스와 상기 버텍스 타일링에 따른 희소 가중 인접 매트릭스 간의 연산을 수행하는 연산부;를 포함하되,

상기 버텍스 타일링부는 상기 거친 모핑 단계에서 제1 특정 기준에 해당할 때까지 상기 희소 가중 인접 매트릭스에 대한 타일의 개수를 반복적으로 증가시키는 것을 특징으로 하는 GCN 가속 장치.

### 청구항 2

제1항에 있어서, 상기 피처 슬라이싱부는

상기 피처 슬라이싱을 위한 슬라이스의 개수를 결정하고 해당 개수에 따라 상기 입력 피처 매트릭스를 분할하여 메모리에 순차적으로 저장하는 것을 특징으로 하는 GCN 가속 장치.

### 청구항 3

삭제

### 청구항 4

삭제

### 청구항 5

제1항에 있어서, 상기 버텍스 타일링부는

상기 거친 모핑 단계를 통해 최적의 파티셔닝 결과를 결정하고, 상기 최적의 파티셔닝 결과를 기초로 상기 정교 모핑 단계를 수행하는 것을 특징으로 하는 GCN 가속 장치.

### 청구항 6

제5항에 있어서, 상기 버텍스 타일링부는

상기 정교 모핑 단계에서 수직 블록들에 관한 제1 방향으로 진행하면서 제2 특정 기준에 해당할 때까지 상기 수직 블록들의 수직 열을 반으로 분할하는 제1 단계와, 제2 방향으로 진행하면서 상기 제2 특정 기준에 해당할 때까지 상기 수직 블록들의 인접한 수직 열들을 병합하는 제2 단계를 수행하는 것을 특징으로 하는 GCN 가속 장치.

### 청구항 7

제1항에 있어서, 상기 연산부는

상기 희소 가중 인접 매트릭스의 타일 별로 상기 입력 피처 매트릭스의 슬라이스들과의 연산을 수행하는 것을 특징으로 하는 GCN 가속 장치.

**청구항 8**

제7항에 있어서, 상기 연산부는

상기 희소 가중 인접 매트릭스의 특정 타일에 있는 버텍스 별로 인접하는 버텍스들 각각에 대해 상기 입력 피처 매트릭스의 슬라이스에 있는 피처들과의 연산을 수행하는 것을 특징으로 하는 GCN 가속 장치.

**청구항 9**

희소 가중 인접 매트릭스 및 입력 피처 매트릭스 간의 연산을 수행하는 GCN(Graph Convolutional Networks) 가속 장치에서 수행되는 GCN 가속 방법에 있어서,

상기 입력 피처 매트릭스에 관한 피처 슬라이싱을 수행하는 단계;

상기 희소 가중 인접 매트릭스에 관한 버텍스 타일링을 수행하는 과정에서, 상기 희소 가중 인접 매트릭스를 기초로 거친 모핑(coarse morphing) 단계와 정교 모핑(fine morphing) 단계로 이루어진 자동 타일 모핑을 수행하는 단계; 및

상기 피처 슬라이싱에 따른 입력 피처 매트릭스와 상기 버텍스 타일링에 따른 희소 가중 인접 매트릭스 간의 연산을 수행하는 단계;를 포함하되,

상기 자동 타일 모핑을 수행하는 단계는 상기 거친 모핑 단계에서 제1 특정 기준에 해당할 때까지 상기 희소 가중 인접 매트릭스에 대한 타일의 개수를 반복적으로 증가시키는 단계를 포함하는 것을 특징으로 하는 GCN 가속 방법.

**청구항 10**

제9항에 있어서, 상기 피처 슬라이싱을 수행하는 단계는

상기 피처 슬라이싱을 위한 슬라이스의 개수를 결정하고 해당 개수에 따라 상기 입력 피처 매트릭스를 분할하여 메모리에 순차적으로 저장하는 단계를 포함하는 것을 특징으로 하는 GCN 가속 방법.

**청구항 11**

삭제

**청구항 12**

제9항에 있어서, 상기 연산을 수행하는 단계는

상기 희소 가중 인접 매트릭스의 타일 별로 상기 입력 피처 매트릭스의 슬라이스들과의 연산을 수행하는 단계를 포함하는 것을 특징으로 하는 GCN 가속 방법.

## *발명의 설명*

### *기 술 분 야*

[0001] 본 발명은 GCN 가속기용 캐시 기술에 관한 것으로, 보다 상세하게는 피처 슬라이싱과 자동 타일 모핑을 적용하

여 GCN 가속의 캐시 동작을 크게 개선하고 더 쉽게 조정할 수 있는 기술에 관한 것이다.

## 배 경 기 술

[0003] 다양한 분야에서 CNN(Convolution Neural Network)이 큰 성공을 거둔 후 GCN(Graph Convolutional Network)은 불규칙한 패턴을 보이는 많은 영역에서 사용될 수 있어 DNN의 차세대 기술로 부상하고 있다. GCN은 기존의 많은 과학적인 문제들과 NLP 등의 고전적인 DNN 응용들에 빠르게 확장되고 있다.

[0004] GCN 가속의 주요 문제 중 하나는 캐시 효율성을 저하시키는 불규칙성(irregularity)이다. 이를 해결하기 위해 인기 있는 솔루션 중 하나는 GCN을 분할하는 것이다. 주어진 행렬을 여러 타일로 분할하면 작업 세트(working set)가 해당 타일로 제한되어 더 나은 캐시 동작이 가능할 수 있다.

[0005] 그러나, 기존 GCN 가속기는 그래프 토폴로지를 타일링하는데 중점을 두고 있다. 토폴로지 타일링은 기존 그래프 처리에 효과적인 접근 방식이었지만 GCN을 사용하면 그래프 토폴로지 데이터와 특징 데이터 간의 균형이 다르기 때문에 장점이 크지 않을 수 있다. 즉, 버텍스 타일링에는 성능 향상 감소와 타일링 매개변수 조정의 어려움이라는 두 가지 문제가 존재한다.

## 선행기술문헌

### 특허문헌

[0007] (특허문헌 0001) 한국공개특허 제10-2018-0123846호 (2018.11.20)

## 발명의 내용

### 해결하려는 과제

[0008] 본 발명의 일 실시예는 GCN의 경우 불규칙적인 데이터 접근으로 인해 연산 과정에서 효율적인 캐시 메모리 사용이 어렵다는 문제를 해결하기 위한 기법으로서 피처 슬라이싱을 활용하여 행렬을 분할하여 계산하고 반복적인 데이터 접근을 타일 모핑을 통해 최적화하여 캐시 메모리 접근을 최대화하는 기술을 제공하고자 한다.

### 과제의 해결 수단

[0010] 실시예들 중에서, 피처 슬라이싱과 자동 타일 모핑을 이용한 GCN 가속 장치는 희소 가중 인접 매트릭스 및 입력 피처 매트릭스 간의 연산을 수행하는 GCN(Graph Convolutional Networks) 가속 장치에 있어서, 상기 입력 피처 매트릭스에 관한 피처 슬라이싱을 수행하는 피처 슬라이싱부; 상기 희소 가중 인접 매트릭스에 관한 버텍스 타일링을 수행하는 버텍스 타일링부; 및 상기 피처 슬라이싱에 따른 입력 피처 매트릭스와 상기 버텍스 타일링에 따른 희소 가중 인접 매트릭스 간의 연산을 수행하는 연산부;를 포함한다.

[0011] 상기 피처 슬라이싱부는 상기 피처 슬라이싱을 위한 슬라이스의 개수를 결정하고 해당 개수에 따라 상기 입력 피처 매트릭스를 분할하여 메모리에 순차적으로 저장할 수 있다.

[0012] 상기 버텍스 타일링부는 상기 희소 가중 인접 매트릭스를 기초로 거친 모핑(coarse morphing) 단계와 정교 모핑(fine morphing) 단계로 이루어진 자동 타일 모핑을 수행할 수 있다.

[0013] 상기 버텍스 타일링부는 상기 거친 모핑 단계에서 제1 특정 기준에 해당할 때까지 상기 희소 가중 인접 매트릭스에 대한 타일의 개수를 2배로 증가시킬 수 있다.

[0014] 상기 버텍스 타일링부는 상기 거친 모핑 단계를 통해 최적의 파티셔닝 결과를 결정하고, 상기 최적의 파티셔닝 결과를 기초로 상기 정교 모핑 단계를 수행할 수 있다.

[0015] 상기 버텍스 타일링부는 상기 정교 모핑 단계에서 상기 수직 블록들에 관한 제1 방향으로 진행하면서 상기 제2 특정 기준에 해당할 때까지 상기 수직 블록들의 수직 열을 반으로 분할하는 제1 단계와, 제2 방향으로 진행하면서 상기 제2 특정 기준에 해당할 때까지 상기 수직 블록들의 인접한 수직 열들을 병합하는 제2 단계를 수행할 수 있다.

[0016] 상기 연산부는 상기 희소 가중 인접 매트릭스의 타일 별로 상기 입력 피처 매트릭스의 슬라이스들과의 연산을

수행할 수 있다.

[0017] 상기 연산부는 상기 희소 가중 인접 매트릭스의 특정 타일에 있는 버텍스 별로 인접하는 버텍스들 각각에 대해 상기 입력 피처 매트릭스의 슬라이스에 있는 피처들과의 연산을 수행할 수 있다.

[0018] 실시예들 중에서, 희소 가중 인접 매트릭스 및 입력 피처 매트릭스 간의 연산을 수행하는 GCN(Graph Convolutional Networks) 가속 장치에서 수행되는 GCN 가속 방법은 상기 입력 피처 매트릭스에 관한 피처 슬라이싱을 수행하는 단계; 상기 희소 가중 인접 매트릭스에 관한 버텍스 타일링을 수행하는 단계; 및 상기 피처 슬라이싱에 따른 입력 피처 매트릭스와 상기 버텍스 타일링에 따른 희소 가중 인접 매트릭스 간의 연산을 수행하는 단계;를 포함한다.

[0019] 상기 피처 슬라이싱을 수행하는 단계는 상기 피처 슬라이싱을 위한 슬라이스의 개수를 결정하고 해당 개수에 따라 상기 입력 피처 매트릭스를 분할하여 메모리에 순차적으로 저장하는 단계를 포함할 수 있다.

[0020] 상기 버텍스 타일링을 수행하는 단계는 상기 희소 가중 인접 매트릭스를 기초로 거친 모핑(coarse morphing) 단계와 정교 모핑(fine morphing) 단계로 이루어진 자동 타일 모핑을 수행하는 단계를 포함할 수 있다.

[0021] 상기 연산을 수행하는 단계는 상기 희소 가중 인접 매트릭스의 타일 별로 상기 입력 피처 매트릭스의 슬라이스들과의 연산을 수행하는 단계를 포함할 수 있다.

## *발명의 효과*

[0023] 개시된 기술은 다음의 효과를 가질 수 있다. 다만, 특정 실시예가 다음의 효과를 전부 포함하여야 한다거나 다음의 효과만을 포함하여야 한다는 의미는 아니므로, 개시된 기술의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.

[0024] 본 발명의 일 실시예에 따른 피처 슬라이싱과 자동 타일 모핑을 이용한 GCN 가속 장치 및 방법은 피처 슬라이싱이 적용하여 GCN 가속기의 캐시 동작을 개선할 수 있고, 피처 슬라이싱이 정확히 동일한 패턴을 여러 번 반복한다는 사실을 기초로 버텍스 타일링 과정에 자동 타일 모핑을 적용함으로써 버텍스 타일링에서의 수동적인 조정의 어려움을 해결할 수 있다.

## *도면의 간단한 설명*

[0026] 도 1은 본 발명에 따른 GCN 가속 장치의 기능적 구성을 설명하는 도면이다.

도 2는 본 발명에 따른 GCN 가속 방법의 일 실시예를 설명하는 순서도이다.

도 3은 GCN 가속기 구조를 설명하는 도면이다.

도 4는 도 3의 GCN 가속기의 처리 과정과 버텍스 타일링 과정을 설명하는 도면이다.

도 5는 본 발명에 따른 버텍스 타일링 과정에서 메모리 액세스를 설명하는 도면이다.

도 6은 본 발명에 따른 피처 슬라이싱 과정의 실시예들을 설명하는 도면이다.

도 7은 본 발명에 따른 자동 타일 모핑의 과정을 설명하는 도면이다.

도 8 및 9는 본 발명에 따른 GCN 가속 방법의 성능 비교 결과를 설명하는 도면이다.

도 10은 본 발명에 따른 미스율과 메모리 액세스에 관한 실험 결과를 설명하는 도면이다.

도 11은 본 발명에 따른 성능 민감도를 설명하는 도면이다.

도 12는 본 발명에 따른 자동 타일 모핑의 반복에 따른 결과를 설명하는 도면이다.

## *발명을 실시하기 위한 구체적인 내용*

[0027] 본 발명에 관한 설명은 구조적 내지 기능적 설명을 위한 실시예에 불과하므로, 본 발명의 권리범위는 본문에 설명된 실시예에 의하여 제한되는 것으로 해석되어서는 아니 된다. 즉, 실시예는 다양한 변경이 가능하고 여러 가지 형태를 가질 수 있으므로 본 발명의 권리범위는 기술적 사상을 실현할 수 있는 균등물들을 포함하는 것으로 이해되어야 한다. 또한, 본 발명에서 제시된 목적 또는 효과는 특정 실시예가 이를 전부 포함하여야 한다거나 그러한 효과만을 포함하여야 한다는 의미는 아니므로, 본 발명의 권리범위는 이에 의하여 제한되는 것으로 이해

되어서는 아니 될 것이다.

[0028] 한편, 본 출원에서 서술되는 용어의 의미는 다음과 같이 이해되어야 할 것이다.

[0029] "제1", "제2" 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위한 것으로, 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.

[0030] 어떤 구성요소가 다른 구성요소에 "연결되어"있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결될 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어"있다고 언급된 때에는 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 한편, 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0031] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함하다"또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이며, 하나 또는 그 이상의 다른 특징이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0032] 각 단계들에 있어 식별부호(예를 들어, a, b, c 등)는 설명의 편의를 위하여 사용되는 것으로 식별부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않는 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0033] 본 발명은 컴퓨터가 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현될 수 있고, 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광 데이터 저장 장치 등이 있다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

[0034] 여기서 사용되는 모든 용어들은 다르게 정의되지 않는 한, 본 발명이 속하는 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한 이상적이거나 과도하게 형식적인 의미를 지니는 것으로 해석될 수 없다.

[0036] 도 1은 본 발명에 따른 GCN 가속 장치의 기능적 구성을 설명하는 도면이다.

[0037] 도 1을 참조하면, GCN 가속 장치(100)는 본 발명에 따른 피처 슬라이싱과 자동 타일 모핑을 이용한 GCN 가속 방법을 수행하는 복수의 기능적 구성들을 포함하여 구현될 수 있다. 즉, GCN 가속 장치(100)는 피처 슬라이싱부(110), 버텍스 타일링부(120) 및 연산부(130)를 포함할 수 있다.

[0038] 피처 슬라이싱부(110)는 입력 피처 매트릭스에 관한 피처 슬라이싱(feature slicing)을 수행할 수 있다. 여기에서, 입력 피처 매트릭스(input)는 GCN(Graph Convolution Network)의 특정 레이어(layer)에 관한 추론(inference) 과정에서의 입력에 해당할 수 있다. 입력 피처 매트릭스의 크기는 그래프에 포함된 노드들의 개수와 각 노드에 저장되는 입력 피처의 차원의 크기에 따라 결정될 수 있다.

[0039] 또한, 특정 레이어에서 입력 피처 매트릭스에 대한 추론을 통해 다음 레이어를 위한 출력 피처 매트릭스(output feature matirx)가 생성될 수 있다. 피처 슬라이싱부(110)는 GCN의 추론 과정에서 수행되는 집계(aggregation) 단계의 연산을 위해 입력 피처 매트릭스를 소정의 슬라이스들로 분할하고 각 슬라이스에 대한 정보를 획득하는 동작을 수행할 수 있다.

[0040] 일 실시예에서, 피처 슬라이싱부(110)는 피처 슬라이싱을 위한 슬라이스의 개수를 결정하고 해당 개수에 따라 입력 피처 매트릭스를 분할하여 메모리에 순차적으로 저장할 수 있다. 이때, 분할되는 슬라이스의 개수는 동작 환경이나 조건에 따라 가변적으로 적용될 수 있으며, 피처 슬라이싱부(110)는 슬라이스의 개수를 결정한 다음 해당 개수에 따라 입력 피처 매트릭스를 수직으로 분할할 수 있다. 예를 들어, m×m의 크기를 갖는 입력 피처 매트릭스를 2개의 슬라이스들로 수직 분할하는 경우 분할된 슬라이스는 m×m/2의 크기로 형성될 수 있다. 또한, 피처 슬라이싱부(110)는 분할된 슬라이스를 메모리 상에 저장하여 보관할 수 있으며, 각 슬라이스의 성분

(element)들은 순차적인 방법(sequential manner)으로 정렬되어 저장될 수 있다.

[0041] 버텍스 타일링부(120)는 희소 가중 인접 매트릭스에 관한 버텍스 타일링을 수행할 수 있다. 여기에서, 희소 가중 인접 매트릭스(sparse weighted adjacency matrix)는 GCN(Graph Convolution Network)의 특정 레이어(layer)에서 그래프 구조에 대한 정보를 포함하는 인접 행렬에 해당할 수 있다. 희소 가중 인접 매트릭스의 크기는 그래프에 포함된 노드들의 개수에 따라 결정될 수 있다. 즉, 희소 가중 인접 매트릭스는 특정 레이어의 추론 과정에서 입력 피처 매트릭스와의 연산에 사용될 수 있다. 버텍스 타일링부(120)는 GCN의 추론 과정에서 수행되는 집계(aggregation) 단계의 연산을 위해 희소 가중 인접 매트릭스를 소정의 타일(tile)들로 분할하고 각 타일에 대한 정보를 획득하는 동작을 수행할 수 있다.

[0042] 일 실시예에서, 버텍스 타일링부(120)는 희소 가중 인접 매트릭스를 기초로 거친 모핑(coarse morphing) 단계와 정교 모핑(fine morphing) 단계로 이루어진 자동 타일 모핑을 수행할 수 있다. 여기에서, 자동 타일 모핑(automatic tile morphing)은 버텍스 타일링 과정에서 최적의 타일 개수를 결정하기 위한 과정에 해당할 수 있다. 즉, 자동 타일 모핑은 피처 슬라이싱부(110)에 의해 수행되는 입력 피처 매트릭스에 대한 피처 슬라이싱과 함께 사용되기 위한 전제 조건으로써 최적화된 타일링(또는 파티셔닝) 결과를 도출하기 위한 방법에 해당할 수 있으며, 크게 두가지 단계를 통해 수행될 수 있다.

[0043] 일 실시예에서, 버텍스 타일링부(120)는 거친 모핑 단계에서 제1 특정 기준에 해당할 때까지 희소 가중 인접 매트릭스에 대한 타일의 개수를 2배로 증가시킬 수 있다. 즉, 자동 타일 모핑의 거친 모핑(coarse morphing) 단계에서는 기 설정된 조건을 충족할 때까지 반복적으로 타일의 개수를 2배로 증가시키는 동작이 수행될 수 있다. 이때, 기 설정된 조건인 제1 특정 기준은 수행 성능이 감소되는 조건에 해당할 수 있으며, 수행 성능은 기 정의된 실행 시간을 기초로 측정될 수 있다. 이에 따라, 거친 모핑 단계에서는 성능이 감소될 때까지 반복적으로 타일의 개수를 2배로 증가시키는 동작이 수행되며, 그 결과 한 번의 반복 단계를 통해 이전보다 4배로 증가된 타일들이 생성될 수 있다.

[0044] 일 실시예에서, 버텍스 타일링부(120)는 거친 모핑 단계를 통해 최적의 파티셔닝 결과를 결정하고, 최적의 파티셔닝 결과를 기초로 정교 모핑 단계를 수행할 수 있다. 거친 모핑 단계는 희소 가중 인접 매트릭스에 대한 최적의 파티셔닝 결과를 도출하는 단계에 해당하며, 최적의 파티셔닝 결과는 반복 과정에서 성능이 최초로 감소되는 시점을 기준으로 이전 단계의 타일링 결과에 해당할 수 있다. 거친 모핑 단계 이후 수행되는 정교 모핑 단계는 분할된 타일들에 대해 미세한 조정을 수행하는 단계에 해당할 수 있다. 정교 모핑 단계를 통해 획득된 최종적인 파티셔닝 결과를 기초로 이후 입력 피처 매트릭스와 희소 가중 인접 매트릭스 간의 연산 동작이 수행될 수 있다.

[0045] 일 실시예에서, 버텍스 타일링부(120)는 정교 모핑 단계에서 수직 블록들에 관한 제1 방향으로 진행하면서 제2 특정 기준에 해당할 때까지 수직 블록들의 수직 열을 반으로 분할하는 제1 단계와, 제2 방향으로 진행하면서 제2 특정 기준에 해당할 때까지 수직 블록들의 인접한 수직 열들을 병합하는 제2 단계를 수행할 수 있다. 여기에서, 제1 방향은 수직 블록들 중 적중률이 가장 낮은 블록을 기준으로 시작하여 적중률이 증가하는 방향에 해당할 수 있으며, 제2 방향은 제1 방향의 역방향에 해당할 수 있다.

[0046] 보다 구체적으로, 정교 모핑 단계에서는 2개의 단계들이 순차적으로 수행될 수 있다. 제1 단계는 적중률이 증가하는 방향으로 이동하면서 수직 방향으로 정렬된 각 블록들을 수평으로 분할하는 반복 과정에 해당할 수 있다. 제2 단계는 적중률이 감소하는 방향으로 이동하면서 수직 방향으로 정렬된 각 블록들을 수평으로 인접한 블록과 병합하는 반복 과정에 해당할 수 있다. 한편, 버텍스 타일링 과정에서 생성되는 타일의 형태를 정사각형으로만 제한하는 경우, 제2 단계에서 수평으로 인접한 블록과의 병합 후 타일의 정사각형 조건을 충족하기 위하여 수직으로 인접한 블록과의 병합 동작이 추가로 수행될 수 있다.

[0047] 연산부(130)는 피처 슬라이싱에 따른 입력 피처 매트릭스와 버텍스 타일링에 따른 희소 가중 인접 매트릭스 간의 연산을 수행할 수 있다. 입력 피처 매트릭스와 희소 가중 인접 매트릭스 간의 연산은 GCN의 특정 레이어에 대한 추론 과정에서 수행될 수 있으며, 해당 과정에서 GCN 가속기의 캐시 동작을 개선하기 위해 피처 슬라이싱과 버텍스 타일링이 적용될 수 있다. 이때, 수행되는 연산은 매트릭스 간의 곱셈 연산에 해당할 수 있다. 또한, 여기에서는 피처 슬라이싱과 버텍스 타일링을 병합 적용하는 것을 예로 들어 설명하고 있으나, 반드시 이에 한정되지 않고, 피처 슬라이싱과 버텍스 타일링이 선택적으로 적용될 수 있음은 물론이다. 예를 들어, 피처 슬라이싱을 단독으로 사용하는 경우 또는 버텍스 타일링과 함께 사용하는 경우에는 기존의 버텍스 타일링을 단독 사용하는 경우보다 더 뛰어난 성능을 달성할 수 있다.

[0048] 일 실시예에서, 연산부(130)는 희소 가중 인접 매트릭스의 타일 별로 입력 피처 매트릭스의 슬라이스들과의 연산을 수행할 수 있다. 보다 구체적으로, 연산부(130)는 희소 가중 인접 매트릭스의 특정 타일(tile)에 있는 버텍스(vertex) 별로 인접하는 버텍스들 각각에 대해 입력 피처 매트릭스의 슬라이스에 있는 피처들과의 연산을 수행할 수 있다. 이에 대해서는 도 6을 통해 보다 자세히 설명한다.

[0049] 제어부(도 1에 미도시함)는 GCN 가속 장치(100)의 전체적인 동작을 제어하고, 피처 슬라이싱부(110), 버텍스 타일링부(120) 및 연산부(130) 간의 제어 흐름 또는 데이터 흐름을 관리할 수 있다.

[0051] 도 2는 본 발명에 따른 GCN 가속 방법의 일 실시예를 설명하는 순서도이다.

[0052] 도 2를 참조하면, GCN 가속 장치(100)는 피처 슬라이싱부(110)를 통해 입력 피처 매트릭스(X)에 관한 피처 슬라이싱을 수행할 수 있다(단계 S230). 이를 위해, GCN 가속 장치(100)는 GCN의 특정 레이어에 관한 입력 피처 매트릭스를 수신할 수 있다(단계 S210).

[0053] 또한, GCN 가속 장치(100)는 버텍스 타일링부(120)를 통해 희소 가중 인접 매트릭스(A)에 관한 버텍스 타일링을 수행할 수 있다(단계 S240). 이를 위해, GCN 가속 장치(100)는 GCN의 특정 레이어에 관한 희소 가중 인접 매트릭스를 수신할 수 있다(단계 S220).

[0054] 또한, GCN 가속 장치(100)는 연산부(130)를 통해 피처 슬라이싱에 따른 입력 피처 매트릭스와 상기 버텍스 타일링에 따른 희소 가중 인접 매트릭스 간의 연산을 수행할 수 있다(단계 S250).

[0056] 이하, 도 3 내지 12를 참조하여 본 발명에 따른 GCN 가속 장치 및 방법에 대해 보다 자세히 설명한다.

[0057] 도 3은 GCN 가속기 구조를 설명하는 도면이고, 도 4는 도 3의 GCN 가속기의 처리 과정과 버텍스 타일링 과정을 설명하는 도면이며, 도 5는 본 발명에 따른 버텍스 타일링 과정에서 메모리 액세스를 설명하는 도면이고, 도 6은 본 발명에 따른 피처 슬라이싱 과정의 실시예들을 설명하는 도면이고, 도 7은 본 발명에 따른 자동 타일 모핑의 과정을 설명하는 도면이며, 도 8 및 9는 본 발명에 따른 GCN 가속 방법의 성능 비교 결과를 설명하는 도면이고, 도 10은 본 발명에 따른 미스율과 메모리 액세스에 관한 실험 결과를 설명하는 도면이며, 도 11은 본 발명에 따른 성능 민감도를 설명하는 도면이고, 도 12는 본 발명에 따른 자동 타일 모핑의 반복에 따른 결과를 설명하는 도면이다.

[0059] 도 3을 참조하면, GCN 가속기의 구조는 조합 엔진(Combination Engine)(310) 및 집계 엔진(Aggregation Engine)(320)을 포함할 수 있다. 일 실시예에서, 조합 엔진(310)은 1GHz로 동작하는 32×32 시스톨릭 배열(systolic array) 모듈(313)로 구성될 수 있으며, 속성 버퍼(Property Buffer) 모듈(311)을 통해 피처(X)를 읽고, 가중치 리더 모듈(312)을 통해 프리페치된 가중치(W)를 읽어 32×32 시스톨릭 배열 모듈(313)을 통과해 X×W 연산을 수행할 수 있다.

[0060] 일 실시예에서, 집계 엔진(320)은 16개의 SIMD 코어들로 구성된 SIMD 코어 모듈(324)을 포함할 수 있으며, 버텍스 프리페치 모듈(321) 및 엣지 프리페치 모듈(322)을 통해 그래프 데이터(A)를 읽고, 피처 리더 모듈(323)을 통해 피처를 읽어 SIMD 코어 모듈(324)에서 A×X 연산을 수행할 수 있다.

[0061] 본 발명에 따른 GCN 가속 장치(100)는 상기와 같은 구조에서 효율적인 연산을 위하여 피처 매트릭스(X)를 분할한 뒤 각각의 분할 매트릭스를 순차적으로 처리할 수 있다. 이때, 한 분할 매트릭스를 처리할 때마다 전체 토폴로지 매트릭스(A)는 반복적으로 읽혀질 수 있다. GCN 가속 장치(100)는 해당 방법을 통해 캐시가 저장해야 하는 작업 세트의 크기(working set size)를 줄일 수 있고, 적중률(hit ratio)를 증가시켜 큰 성능 향상을 제공할 수 있다. 또한, GCN 가속 장치(100)는 매 반복마다 타일 모핑(tile morphing)을 추가 적용함으로써 버텍스 타일링의 효과도 얻을 수 있는 최적의 근접한 패턴을 빠르게 찾을 수 있어 수동적으로 찾은 최적값과 근사한 효과를 제공할 수 있다.

[0062] 한편, GCN에서 레이어 $^{l}$의 추론(inference)은 다음의 수학식 1과 같이 표현될 수 있다.

[0063] [수학식 1]

[0064] $$X^{(l+1)} = \sigma\left(\tilde{A} \cdot X^{(l)} \cdot W^{(l)}\right)$$

[0066] 여기에서, X는 해당 레이어의 입력 피처 매트릭스이고, $\tilde{A}$는 CSC(Compressed Sparse Column) 형식의 표현된 그래프의 희소 가중 인접 매트릭스이며, W는 가중치 매트릭스(weight matrix)이고, σ는 비선형 함수(non-

linearity function)(예를 들어, ReLU)이다.

[0067] 또한, $\tilde{A} \cdot X^{(l)}$와 그 변형들은 집계(aggregation)에 해당할 수 있고, $X \cdot W$의 변형들은 조합(combination)에 해당할 수 있다. 한편, 기존의 많은 가속기(accelerator)들은 집계과 조합의 고유한 계산 특성으로 인해 SIMD 유닛들과 시스톨릭 배열(systolic array)들에 관한 하이브리드 구조(hybrid architecture)를 채택하고 있다.

[0068] 도 3의 경우, 기본적인 하이브리드 GCN 가속기(300)에 해당할 수 있다. 버텍스 프리페치(Vertex Prefetch) 모듈(321)은 CSC의 행 인덱스들을 읽고 이를 엣지 프리페치(Edge Prefetch) 모듈(322)에 중계하여 엣지 정보를 피처 리더(Feature Reader) 모듈(323)에 제공할 수 있다. 그런 다음, 피처 리더 모듈(323)은 엣지들의 소스(source)로부터 버텍스 피처(vertex feature)들을 수집할 수 있다. 소스 버텍스(source vertex)들은 매우 랜덤화된 액세스 패턴(randomized access pattern)을 보이기 때문에 집합 연관 캐시(set associative cache)로서 구성된 대규모 전역 메모리(large global memory)가 사용될 수 있다. 마지막으로, 버텍스 피처들이 계산될 수 있고 SIMD 코어 모듈(324)에 의해 전역 캐시(global cache)에 누적될 수 있다. 그런 다음, 버텍스 피처들은 조합 엔진(310)에 의해 처리되어 다음 GCN 레이어의 입력 피처(input feature)가 될 수 있다.

[0070] 도 4를 참조하면, GCN의 두 단계 중 집계 단계(aggregation phase)는 빈번한 랜덤 메모리 액세스와 낮은 캐시 적중률(hit ratio)로 인해 그림 (a)와 같이 종종 성능 병목 현상(performance bottleneck)이 발생할 수 있다. 이러한 문제를 완화하기 위해, 버텍스 타일링(vertex tiling)은 피처들을 캐시에 맞출 수 있는 좋은 방법에 해당할 수 있다.

[0071] 그림 (b)는 GCN의 레이어 $^{l}$에 대해 기존의 버텍스 타일링을 실행하는 과정에 해당할 수 있다. 타일 그룹 T를 형성하기 위해 차원당 2개의 타일들이 존재할 수 있다(즉, $B_V = 2$). $\tilde{A}$의 컬러 스트립(colored strip)들은 엣지들에 해당할 수 있고, X의 수평 스트립(horizontal strip)들은 매칭되는 컬러 엣지(colored edge)들로부터 수집되는 피처들에 해당할 수 있다. 가장 바깥쪽 루프 ①에서는 T에서 버텍스들의 타일들이 방문될 수 있다. 각 타일 내에서 행(row)들(버텍스들)은 ②에서 액세스될 수 있다. ③에서 만약 버텍스가 타일 내에서 엣지를 갖는 경우, ④에서 대응되는 피처들이 수집될 수 있다. 오직 타일 내의 피처들만이 액세스되기 때문에(도 4의 '작업 집합'), 높은 캐시 적중률(hit ratio)이 유지될 수 있다. 그러나, 이 경우 출력 피처 매트릭스에 여러 번 액세스해야 하는 비용을 초래할 수 있다.

[0073] 도 5를 참조하면, 버텍스 타일링은 입력 피처들의 적중률(hit rate)을 높이는데 효과적이지만, 출력 피처 매트릭스에 여러 번 액세스 하는 것이 요구될 수 있다. 이러한 요구는 피처 너비(feature width)가 작을 때에는 큰 문제가 아닐 수 있지만, GCN의 경우 반복 횟수가 증가할수록 액세스 비용에 큰 영향을 미칠 수 있다. 도 5의 경우, 소셜 네트워크 그래프(social network graph)에 대한 버텍스 타일링의 메모리 액세스 횟수를 나타낼 수 있다. X축은 $|B_V|$이고, Y축은 메모리 액세스의 양(volume)에 해당할 수 있다. 64×64 타일들에서 증가하는 적중률에 의해 입력 피처들에 대한 액세스 비용이 충분히 감소하는 동안, 출력 피처들에 대한 반복이 이미 액세스 비용을 지배하기 시작하여 속도 향상에 대한 이득을 상쇄시킬 수 있다.

[0075] 도 6을 참조하면, 도 5에서의 관찰을 기반으로, 피처 슬라이싱은 피처 매트릭스를 토폴로지(topology) 대신 열(column)들로 분할할 수 있다. 이러한 동작은 작업 세트(working set)의 크기를 줄이는 목적으로 사용될 수 있으나, 다른 트레이드오프(trade-off)를 발생시킬 수 있다.

[0076] 그림 (a)는 2개의 슬라이스들(즉, $B_F = 2$)에 관한 피처 슬라이싱의 과정을 나타낼 수 있다. 피처 슬라이싱은 가장 바깥쪽 루프 ①에서 피처 매트릭스의 수평으로 분할된 2개의 슬라이스들을 방문할 수 있다. 그런 다음, 피처 슬라이싱은 ②에서 버텍스들, ③에서 엣지들, 그리고 ④에서 피처들에 대해 수행될 수 있다. 이러한 방식은 다중 그래프 순회(multiple graph traversal)를 포함하는 경우 비효율적일 수 있지만, GCN의 처리 속도를 크게 향상시킬 수 있다.

[0077] 트레이드오프를 보다 구체적으로 설명하기 위해, 메모리 액세스 횟수(memory access count)에 대한 1차 모델(first-order model)이 도출될 수 있다. 미스율 m(x)가 작업 세트의 크기 x에 관한 함수라고 대략적으로 가정하면, |V| 버텍스들, |E| 엣지들, |F| 버텍스당 피처들 및 $B_V \times B_V$ 타일들에 관한 버텍스 타일링의 메모리 액세스 횟수 $M_{VT}$는 다음의 수학식 2와 같이 표현될 수 있다.

[0078] [수학식 2]

[0079] $$M_{VT} = (B_V|V| + |E|) + m(|V||F|/B_V) \cdot |E||F| + B_V|V||F|$$

[0081] 여기에서, 첫 번째 항 $B_V|V|+|E|$는 CSC 포맷에서 그래프 방문(graph traversal)의 메모리 액세스 횟수(memory access count)에 해당한다. 두 번째 항 $m(|V||F|/B_V) \cdot |E||F|$는 입력 피처 액세스들(input feature accesses)에 해당한다. 조밀 매트릭스 곱셈(dense matrix multiplication)에서, $|V| \times |V|$ 매트릭스 및 $|V| \times |F|$ 매트릭스 간의 곱셈은 $|V|^2|F|$에 해당할 수 있다. 그러나, $\tilde{A}$는 희소 행렬(sparse matrix)이므로 행렬 X에서 액세스되는 총 행의 수는 |E|가 될 수 있다. 마지막으로, |V||F|는 출력 피처 액세스들(output feature accesses)에 해당할 수 있다.

[0082] 상기의 수학식 2에서와 같이, 미스율(miss rate) 함수의 매개변수는 $B_V$와 함께 감소함으로써 더 나은 캐시 동작에 기여할 수 있다. 그러나, 동시에 $B_V$는 출력 피처 액세스 항(output feature access term) |V||F| 및 버텍스 액세스 항(vertex access term) |V|에 곱해질 수 있다. 이러한 오버헤드(overhead)는 |F|가 그래프 토폴로지(graph topology)에 비해 작은 경우 크지 않을 수 있다.

[0083] 그러나, GCN의 경우, |F|는 일반적으로 수백에서 수천까지의 범위를 가질 수 있기 때문에, 버텍스 타일링은 매력적이지 않은 방법으로 보일 수 있다. 반면에, 피처 슬라이싱이 $B_F$개의 슬라이스들과 함께 적용되는 경우, 메모리 액세스 횟수 $M_{FS}$는 다음의 수학식 3과 같이 표현될 수 있다.

[0084] [수학식 3]

[0085] $$M_{FS} = B_F(|V| + |E|) + m(|V||F|/B_F) \cdot |E||F| + |V||F|$$

[0087] 상기의 수학식 2와 유사하게, 피처 슬라이싱은 동일한 방식으로 입력 피처 액세스의 양을 $B_F$배로 줄일 수 있다. 차이점은 액세스 반복 횟수(access repetition count)가 첫 번째(그래프 순회) 항에 존재한다는 것일 수 있다.

[0088] 그러나, 그래프 순회의 추가 비용은 피처 액세스 항(feature access term) |V||F|에 의해 작아질 수 있다. 버텍스 타일링과 비교할 때, 동일한 수의 슬라이스/타일(즉, $B_F = B_V = B$)을 가정하면 피처 슬라이싱은 버텍스 타일링보다 다음의 수학식 4 및 5와 같이 표현되는 조건에서 우수한 성능을 나타낼 수 있다.

[0089] [수학식 4]

[0090] $$B|V| + |E| + B|V||F| > B(|V| + |E|) + |V||F|$$

[0091] [수학식 5]

[0092] $$|F| > |E|/|V|$$

[0094] 여기에서, |E|/|V|는 일반적으로 약 10의 범위를 갖는 평균 값(average degree)에 해당한다. |F|가 수백 개 정도이기 때문에, 피처 슬라이싱은 종종 훨씬 더 나은 트레이드오프를 제공할 수 있다.

[0095] 슬라이스의 크기($|F|/B_F$)가 너무 작으면 DRAM에서 더 많은 행 활성화(row activation)가 발생할 수 있다. 이러한 효과를 완화하기 위해, 피처 매트릭스는 각 슬라이스 내의 성분(element)이 메모리 내에 순차적으로 저장되도록 매핑될 수 있다. 이를 통해, 더 나은 행 버퍼 지역성(row buffe r locality)과 DRAM에서 더 많은 대역폭(bandwidth)이 도출될 수 있다.

[0096] 피처 슬라이싱은 일반적으로 효율적인 캐시 동작을 위해 버텍스 타일링보다 더 나을 수 있지만, 두 가지 이유로 인해 두 기술을 함께 사용하는 것이 종종 더 유리할 수 있다. 첫째, 피처 슬라이싱은 슬라이스의 최대 개수(즉, $B_F$)에 제한될 수 있다. $B_F$가 너무 커서 각 슬라이스가 캐시 라인보다 작아지는 경우 메모리 대역폭이 크게 낭비될 수 있으며, 이는 메모리 액세스 횟수를 줄이는 목적을 저해시킬 수 있다. 둘째, $B_F$가 m()을 충분히 낮게 유지하기 위해 너무 커야 하는 경우 반복적인 토폴로지 액세스 횟수(topology access count) $B_F(|V|+|E|)$도 무시할 수 없게 된다. 이러한 경우, 도 6의 그림 (b)와 같이 버텍스 타일링과 피처 슬라이싱이 함께 사용될 수

있다. 작업 집합의 크기는 토폴로지 및 피처 배열에 대한 반복의 어느 한쪽에 너무 많은 부담을 주지 않으면서 충분히 작아질 수 있다.

[0097] 동일한 메모리 비용 모델(memory cost model)을 사용하여, 액세스 횟수(access count) $M_{FV}$는 다음의 수학식 6과 같이 모델링될 수 있다.

[0098] [수학식 6]

[0099] $$M_{FV} = B_F(B_V|V| + |E|) + m(|V||F|/B_F B_V) \cdot |E||F| + B_V|V||F|$$

[0101] 상기의 수학식 6은 피처 슬라이싱 및 버텍스 타일링의 일반화(generalization)에 해당될 수 있다. 여기에서, $B_F$ = 1이면 상기의 수학식 2에 해당되고, $B_V$ = 1이면 상기의 수학식 3에 해당될 수 있다. $B_F$와 $B_V$는 모두 작업 집합의 크기를 줄이는데 기여하는 반면, 승수들(multipliers)은 그래프 액세스 항(graph access term)과 피처 액세스 항(feature access term)으로 분할될 수 있다.

[0102] 버텍스 타일링은 타일들의 개수(number of tiles)에 민감하기 때문에 세심한 조정이 요구될 수 있다. 또한, 최적의 $B_V$는 그래프 토폴로지(graph topology)와 피처 너비(feature width)에 따라 달라질 수 있다. 이는 성능 최적화(performance optimization)에 어려움을 제공할 수 있으며, 피처 슬라이싱과 함께 사용하는 경우에도 문제가 지속될 수 있다.

[0103] 다행히도, 피처 슬라이싱은 그래프가 여러 번 순회되고 각 순회가 정확히 동일한 패턴으로 메모리에 액세스한다는 매우 유용한 속성을 가질 수 있다(도 6의 그림 (b)에서, ②-⑤). 본 발명에 따른 자동 타일 모핑은 이를 활용하여 타일링이 없는 상태에서 시작하여 적은 수의 온라인 반복(on-line iterations)으로 거의 최적의 타일링에 도달할 수 있다.

[0105] 도 7을 참조하면, 본 발명에 따른 자동 타일 모핑 절차는 2가지 단계로 수행될 수 있다. 거친 모핑 단계(Coarse Morphing)에서는 실행 시간이 타일의 개수에 관한 볼록 함수(convex function)라고 가정하고 성능이 감소할 때까지 타일의 개수를 두 배로 늘릴 수 있다. 정교 모핑 단계(Fine Morphing)는 거친 모핑 단계의 최상의 분할(도 7의 경우 4×4)에서 개시될 수 있다. 적중률(hit rate)이 가장 낮은 순서(도 7의 경우 왼쪽에서 오른쪽 방향)에 따라 블록들의 수직 스트립(vertical strip)이 성능이 감소할 때까지 반으로 더 분할될 수 있다. 이후 가장 높은 적중률 순으로(도 7의 경우 오른쪽에서 왼쪽 방향) 성능이 감소할 때까지 인접한 두 열이 병합될 수 있다. 검색 공간의 크기(search space size)를 줄이기 위해 타일링을 다양한 크기를 갖는 정사각형 타일(square tile)들의 수직 스트립들로 제한함으로써 정교 모핑은 1D 분할 문제(1D partitioning proble)로 효과적으로 변환될 수 있다. 구현을 위해 단위 타일의 크기(unit tile size)를 정의하고 각 수직 스트립의 길이를 가장 작은 타일 단위로 저장할 수 있다. 실제로 단위 타일은 전역 캐시(global cache)에 완벽하게 맞는 타일 크기로 설정될 수 있다. 여기에서는 토폴로지를 64×64 단위 타일로 나눌 수 있으나, 반드시 이에 한정되지 않음은 물론이다. 최악의 경우 최대 64 단위-너비(unit-width)를 갖는 스트립들이 존재할 수 있으며, 이 스트립의 너비를 저장하는데 1KB 미만의 메모리가 사용될 수 있다.

[0106] 경험적으로, 자동 타일 모핑은 정적 최선(static best)에 가까운 솔루션을 제공할 수 있다. 그러나, 일부 첫 번째 피처 슬라이스는 차선의 타일링(sub-optimal tiling)으로 실행될 수 있다. $B_F$가 크면 이러한 오버헤드는 반복을 통해 감소될 수 있다. 그러나, $B_F$가 작으면 이 오버헤드의 일부가 상대적으로 커져서 더 높은 $B_F$를 선호하는 또 다른 이유가 될 수 있다.

[0108] 도 8을 참조하면, GCN 추론의 성능 비교 결과를 확인할 수 있다. 보다 구체적으로, 'Baseline'은 타일링이나 슬라이싱을 사용하지 않는 아키텍처에 해당할 수 있다. 'VT only'는 버텍스 타일링에 해당할 수 있다. 기존 가속기들과의 비교를 위해 버텍스 타일링 및 차수 인식 버텍스 캐싱(degree-aware vertex caching)을 사용하는 EnGN과 열 기반 곱연산(column-based product), 출력 버퍼 및 타일링 방법을 사용하는 AWB-GCN의 기술이 사용될 수 있다. 공정한 비교를 위해 모든 baselines에 대해 동일한 양의 온칩 메모리(on-chip memory)와 계산 장치(computation unit)가 적용될 수 있다. 'FS only'는 피처 슬라이싱이 단독으로 사용되는 것을 나타내고, 'FS+VT'는 피처 슬라이싱과 버텍스 타일링이 함께 사용되는 것을 나타낸다. 다양한 $B_F$와 $B_V$에 대해 실험이 수행될 수 있으며, 그 중 최선의 값들이 비교 과정에 사용될 수 있다. 마지막으로, 'FS+VT(ATM)'는 자동 타일 모핑 방식에 해당한다.

[0109] 타일링/슬라이싱이 없는 baseline과 비교한 결과, 피처 슬라이싱만 사용하면 기하 평균(geometric mean)에서 16MB 캐시에 대해 18.9%의 속도 향상을 획득할 수 있다. 버텍스 타일링은 반복되는 피처 매트릭스 액세스가 캐시 적중률의 이득을 상쇄하기 때문에 7.9%의 한계 속도 향상을 획득할 수 있다. 피처 슬라이싱과 버텍스 타일링을 함께 사용하면 작업 세트의 크기가 효율적으로 줄어들고 baseline보다 기하 평균에서 40.1%의 뛰어난 속도 향상을 획득할 수 있다. 8MB의 더 작은 캐시를 사용하면 피처 슬라이싱을 단독으로 사용하는 경우 11.1%, FS+VT의 경우 30.6%의 속도 향상을 나타내어 추세가 비슷할 수 있다.

[0110] 본 발명에 방법은 기존의 다른 작업들을 능가할 수 있다. EnGN은 버텍스 타일링을 기반으로 하며 많은 반복 횟수로 인한 문제를 가질 수 있다. 또한, 이미 높은 적중률을 나타내는 큰 차수의 버텍스들에 대한 캡처(capture)가 너무 많기 때문에 차수 인식 버텍스 캐싱이 속도 향상에 미미한 영향을 미칠 수 있다. AWB-GCN에는 열 기반의 곱연산(column product)을 기반으로 하는 다른 데이터 흐름이 나타날 수 있다. 본 발명에 따른 방법과의 두 가지 주요한 차이가 존재할 수 있다. 첫째, AWB-GCN은 모든 버텍스들에 적합한 크기의 큰 출력 버퍼에 종속적일 수 있다. 몇 MB를 필요로 하는 그래프에서 버텍스가 수백만 개라는 점을 감안하면 그 크기는 무시할 수 없고, 실행 가능한 그래프의 크기는 스필링(spilling)에 대한 별도의 고려없이 제한될 수 있다. 둘째, AWB-GCN은 피처 매트릭스를 분할하지만 그 목적은 피처 배열(feature array)이 아니라 토폴로지 데이터(topology data)를 재사용하는 것일 수 있다. 또한, 버텍스 타일링과 유사하게 피처 배열에 대한 반복 횟수가 추가될 수 있다.

[0111] $B_F$ 크기 제한으로 인해 버텍스 타일링이 Pokec 데이터 세트에 대한 피처 슬라이싱보다 약간 더 나은 성능을 발휘한다는 점은 주목할 가치가 있다. 그러나, FS+VT는, FS만 또는 VT만으로 도달 불가능한, 너무 많은 반복없이 적절한 작업 세트의 크기에 도달할 수 있기 때문에 훨씬 더 빠른 속도 향상을 달성할 수 있다.

[0113] 도 10을 참조하면, 그림 (a)는 Pokec 그래프에서 관찰된 $B_F \times B_V$에 대한 캐시 미스율(miss rate)을 나타낸다. 각 $B_F \times B_V$에 대해 가능한 모든 조합을 실험한 결과 최선의 조합이 도출될 수 있다. VT 및 VT+FS의 미스율 하락은 대부분 유사하며, 비용 모델에 대한 상기의 가정들은 일반적으로 미스율이 대략 작업 세트의 크기에 관한 함수라는 점을 나타낼 수 있다.

[0114] 그림 (b)는 미스율이 성능으로 변환되는 방식을 설명하는 메모리 액세스 횟수에 관한 분석 결과를 나타낸다. 밝은 색상의 막대는 피처 액세스를 나타내고 어두운 색상의 막대는 토폴로지 데이터 액세스를 나타낸다. 버텍스 타일링을 사용하면 출력 피처 액세스에 추가되는 반복 횟수가 미스율 하락으로 인한 이득을 상쇄하고 결과적으로 이득이 감소될 수 있다. 반면에 피처 슬라이싱을 사용하면 반복이 대부분 더 작은 토폴로지 데이터에 추가되기 때문에 미스율 하락은 거의 직접적으로 메모리 액세스 감소로 변환될 수 있다.

[0116] 도 11을 참조하면, BF 및 BV에 대한 FS+VT 방식의 민감도를 나타내며, 버텍스 타일의 개수 $B_V$를 조정하는 것이 어려움을 의미할 수 있다. 일반적으로 $B_F$를 가능한 최대값으로 조정하는 것이 최선의 결과를 제공할 수 있다. $B_F$를 늘리는 것이 일반적으로 $B_V$보다 더 나은 트레이드오프를 제공할 수 있고 $B_F$의 크기는 캐시 라인의 크기에 의해 제한되기 때문일 수 있다. 반면에 $B_V$는 개별적인 그래프 토폴로지에 너무 많이 의존하기 때문에 조정하기가 더 어려울 수 있다.

[0117] 도 8의 FS+VT(ATM)는 자동 타일 모핑이 최적에 가까운 타일링을 자동으로 찾아 이러한 조정의 어려움을 완화하는 방법을 제공할 수 있다. 기하 평균(geometric mean)에서 초기 모핑 반복 동안 차선의 성능(sub-optimal performance)으로 인한 오버헤드를 포함하여 정적으로 선택된 최선의 성능을 97.0% 달성할 수 있다.

[0119] 도 12를 참조하면, Pokec 데이터셋의 자동 타일 모핑 과정의 일 실시예를 나타낼 수 있다. 처음 네 번의 반복은 거친 모핑 단계이며, 점차적으로 $B_V = 1$에서 $B_V = 16$으로 이동하면서 성능이 더 나빠지고 $B_V = 8$에서 안정될 수 있다. 이 시점에서 성능은 이미 최적에 가깝고 $B_V = 1$에서 시작함에 있어 너무 많은 오버헤드가 발생하는 것을 방지할 수 있다. 정교 모핑 단계에서 적중률이 가장 낮은 첫 번째 열을 더 분할하고 적중률이 가장 높은 마지막 두 열을 병합할 수 있다.

[0120] 도 7 및 10에서, 피처 슬라이싱의 속도가 $B_F$에 따라 달라진다는 것이 도출될 수 있다. 이것은 피처 너비가 감소되는 GCN의 이후 레이어에 대한 문제일 수 있다. 도 9는 피처 너비에 대한 민감도를 나타낸다. 피처 너비가 16으로 떨어지면 버텍스 타일링에 대한 속도 향상이 7%로 낮아질 수 있다. 그러나, 여전히 기존 작업보다 성능이 우수하고 레이어의 실행 시간(execution time)은 대략적으로 피처 너비에 비례하기 때문에 전체 GCN에 미치는

영향은 작을 수 있다.

[0122] 결과적으로, 본 발명에 따른 GCN 가속 장치 및 방법은 GCN 가속기의 캐시 동작을 개선하기 위하여 피처 슬라이싱이 적용될 수 있다. 실험 결과에 따르면 피처 슬라이싱을 단독으로 사용하거나 또는 버텍스 타일링과 함께 사용하는 경우 본 발명에 따른 방법은 기존의 버텍스 타일링보다 뛰어난 성능을 제공할 수 있다. 또한, 피처 슬라이싱이 정확히 동일한 패턴을 여러 번 반복한다는 사실을 활용하여 본 발명에 따른 방법은 버텍스 타일링 과정에서 최적에 가까운 솔루션을 자동으로 찾을 수 있는 자동 타일 모핑을 적용함으로써 버텍스 타일링에서의 수동적인 조정의 어려움을 해결할 수 있다.

[0124] 상기에서는 본 발명의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

## 부호의 설명

[0126] 100: GCN 가속 장치

110: 피처 슬라이싱부 120: 버텍스 타일링부

130: 연산부

300: GCN 가속기 310: 조합 엔진

311: 속성 버퍼 모듈 312: 가중치 리더 모듈

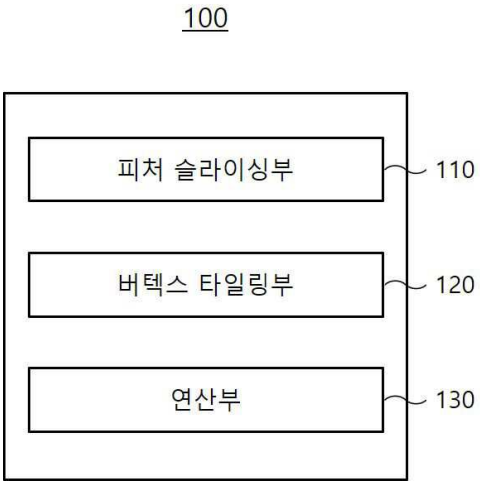
313: 시스톨릭 배열 모듈 320: 집계 엔진

321: 버텍스 프리페치 모듈 322: 엣지 프리페치 모듈

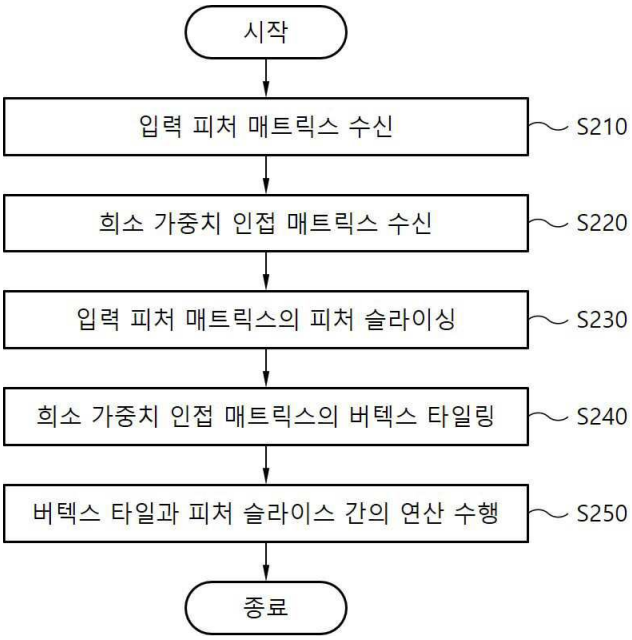
323: 피처 리더 모듈 324: SIMD 코어 모듈

## 도면

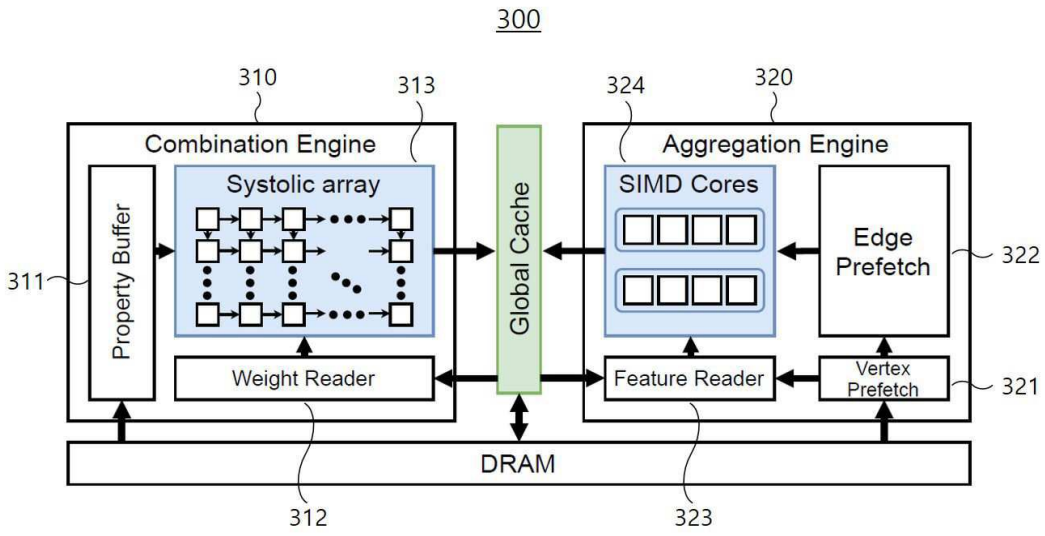
### 도면1



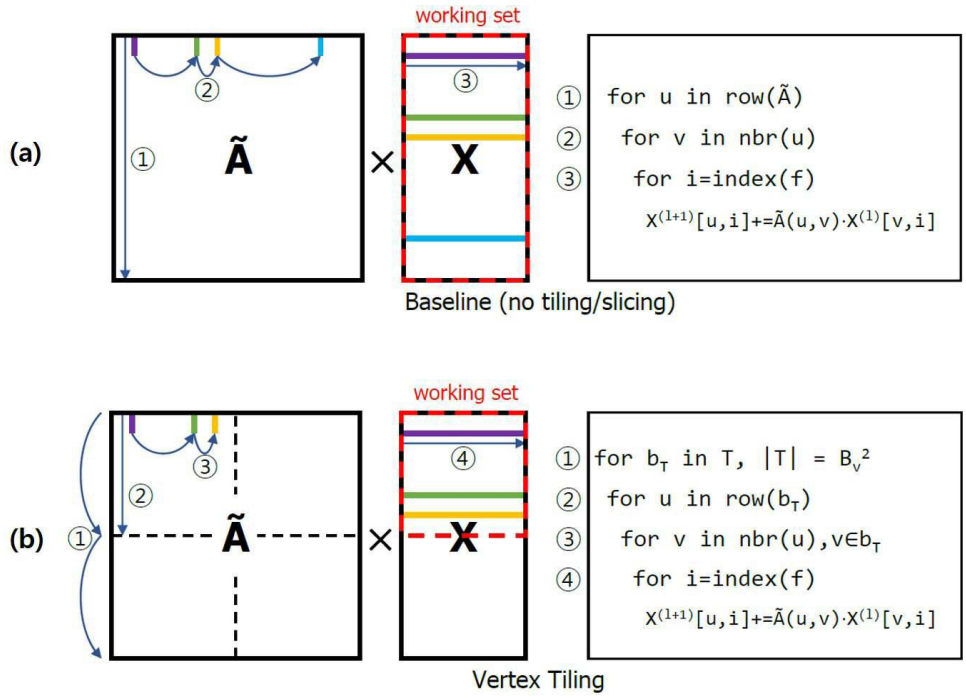
도면2


시작
입력 피처 매트릭스 수신
S210
희소 가중치 인접 매트릭스 수신
S220
입력 피처 매트릭스의 피처 슬라이싱
S230
희소 가중치 인접 매트릭스의 버텍스 타일링
S240
버텍스 타일과 피처 슬라이스 간의 연산 수행
S250
종료

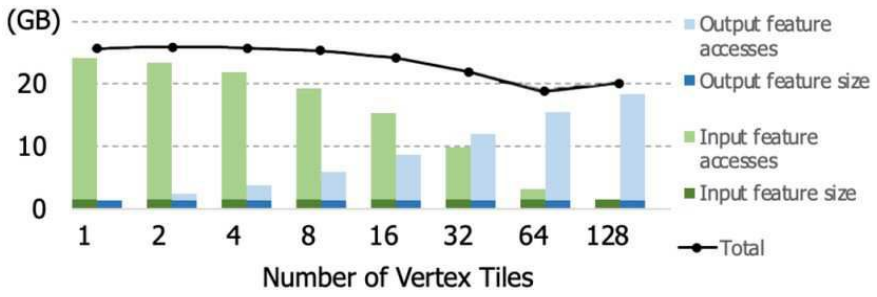
도면3


300
310
313
324
320
Combination Engine
Aggregation Engine
Systolic array
SIMD Cores
Property Buffer
Global Cache
Edge Prefetch
322
311
Weight Reader
Feature Reader
Vertex Prefetch
321
DRAM
312
323

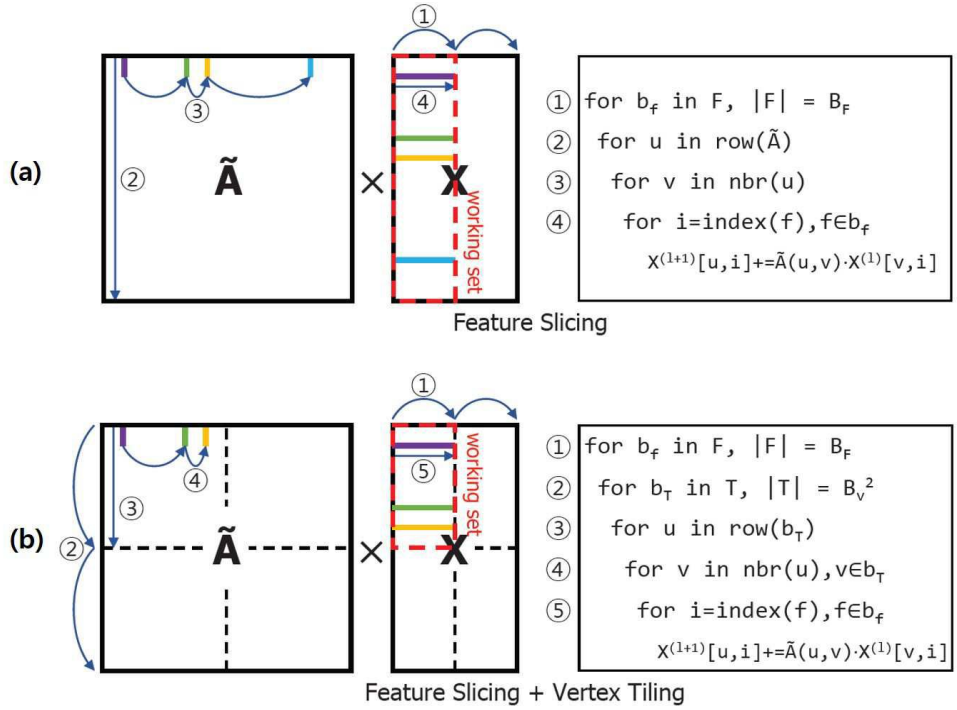
도면4


(a)
working set
Ã
X
① for u in row(Ã)
② for v in nbr(u)
③ for i=index(f)
X(l+1)[u,i]+=Ã(u,v)·X(l)[v,i]
Baseline (no tiling/slicing)
(b)
working set
Ã
X
① for b_T in T, |T| = B_V²
② for u in row(b_T)
③ for v in nbr(u),v∈b_T
④ for i=index(f)
X(l+1)[u,i]+=Ã(u,v)·X(l)[v,i]
Vertex Tiling

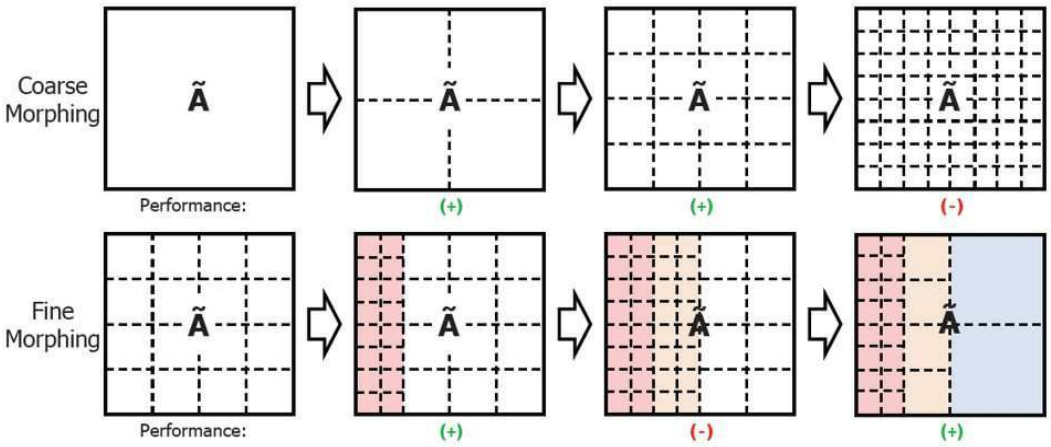
도면5


(GB)
20
10
0
1
2
4
8
16
32
64
128
Number of Vertex Tiles
Output feature accesses
Output feature size
Input feature accesses
Input feature size
Total

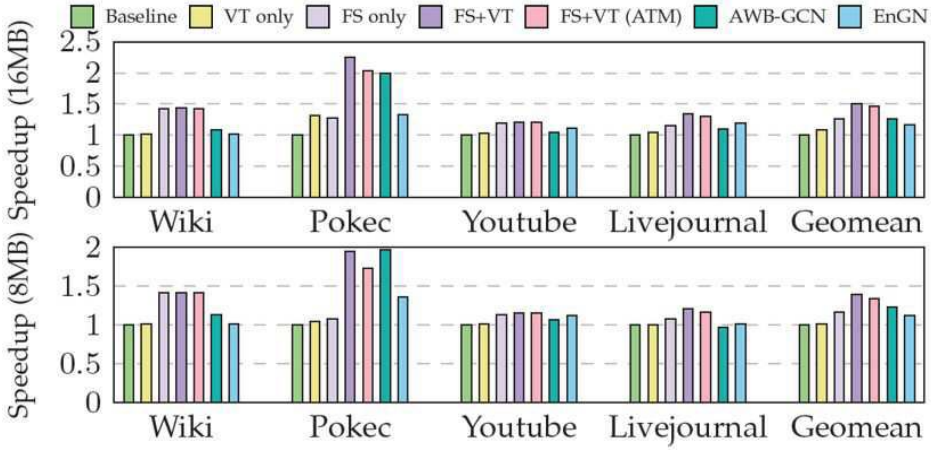
도면6


(a)
Ã
X
working set
① for $b_f$ in F, |F| = $B_F$
② for u in row(Ã)
③ for v in nbr(u)
④ for i=index(f),f∈$b_f$
$X^{(l+1)}[u,i]+=\tilde{A}(u,v)\cdot X^{(l)}[v,i]$
Feature Slicing
(b)
Ã
X
working set
① for $b_f$ in F, |F| = $B_F$
② for $b_T$ in T, |T| = $B_V^2$
③ for u in row($b_T$)
④ for v in nbr(u),v∈$b_T$
⑤ for i=index(f),f∈$b_f$
$X^{(l+1)}[u,i]+=\tilde{A}(u,v)\cdot X^{(l)}[v,i]$
Feature Slicing + Vertex Tiling

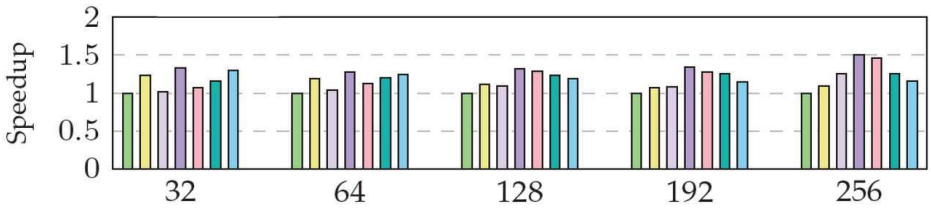
도면7


Coarse Morphing
Performance:
(+)
(+)
(-)
Fine Morphing
Performance:
(+)
(-)
(+)

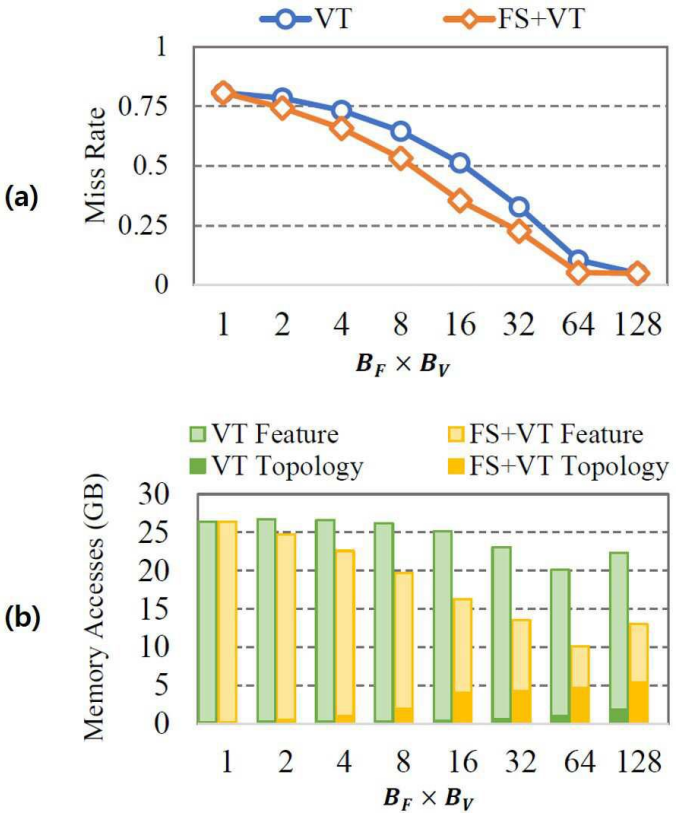
도면8


Baseline
VT only
FS only
FS+VT
FS+VT (ATM)
AWB-GCN
EnGN
Speedup (16MB)
Speedup (8MB)
2.5
2
1.5
1
0.5
0
Wiki
Pokec
Youtube
Livejournal
Geomean

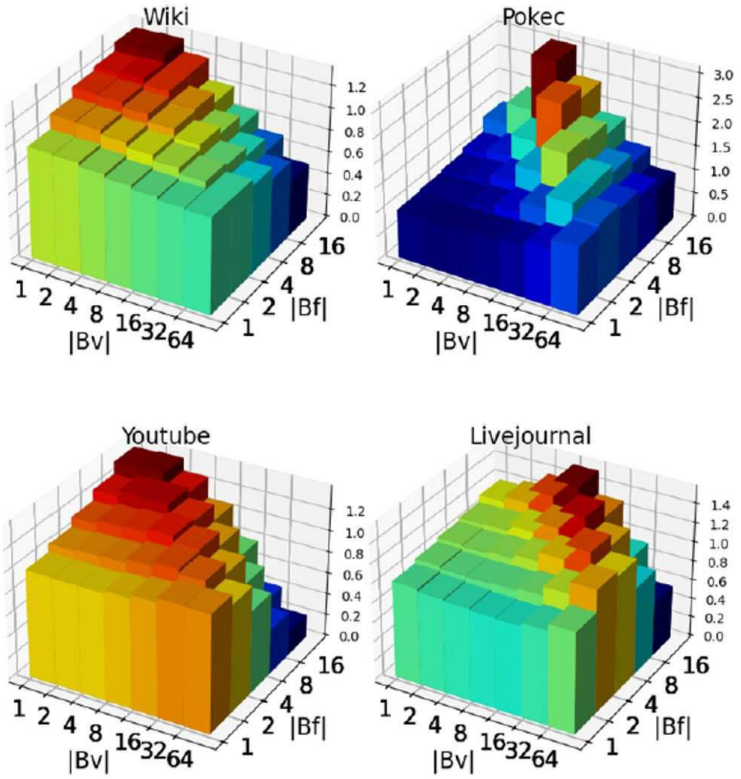
도면9


Speedup
2
1.5
1
0.5
0
32
64
128
192
256

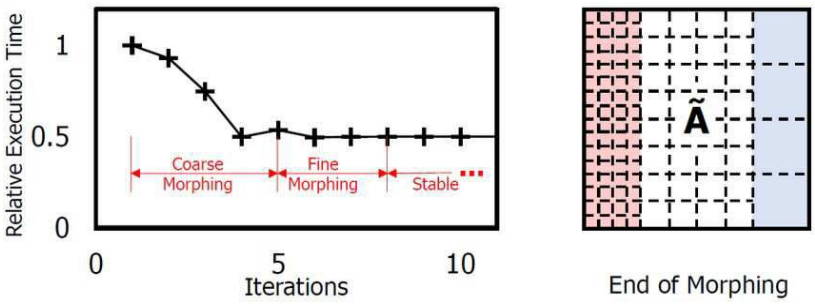
도면10


(a)
VT
FS+VT
Miss Rate
1
0.75
0.5
0.25
0
1
2
4
8
16
32
64
128
$B_F \times B_V$
(b)
VT Feature
VT Topology
FS+VT Feature
FS+VT Topology
Memory Accesses (GB)
30
25
20
15
10
5
0
1
2
4
8
16
32
64
128
$B_F \times B_V$

도면11


Wiki
Pokec
Youtube
Livejournal
|Bv|
|Bf|
1 2 4 8 16 32 64
1 2 4 8 16

도면12


Relative Execution Time
1
0.5
0
0
5
10
Iterations
Coarse Morphing
Fine Morphing
Stable
Ã
End of Morphing