# 희소 행렬-벡터 곱셈 가속기에서 성능 향상을 위한 부하 균형 연구

김광래, 허지현, 정기석*
한양대학교

kksilver91@hanyang.ac.kr, jhheo@hanyang.ac.kr, *kchung@hanyang.ac.kr

# Research on Load Balancing for Performance Improvement in Sparse Matrix-Vector Multiplication Accelerators

Kwang-Rae Kim, Ji-Hyeon Heo, Ki-Seok Chung*
Hanyang University, Seoul, Korea

요 약

희소 행렬-벡터 곱셈 연산 (SpMV)에서는 메모리 성능과 계산 효율성을 높이는 것이 중요하다. 특히 SpMV 는 희소 행렬에 대한 불규칙한 접근 패턴으로 인해 메모리 접근의 성능 저하가 발생한다. 이전 연구는 Processing Element (PE)를 다 수 사용하여 병렬 처리하는 SpMV 가속기를 통해서 이러한 문제를 해결하는 연구를 하였으나, PE 에서의 부하 불균형 문제가 존재하여, PE 활용에 효율성이 떨어져 성능 향상이 제한적이라는 문제가 있었다. 따라서 본 논문에서는 기존의 방법을 적용한 SpMV 가속기에 소수의 PE 를 추가하여 불균형적인 부하를 분담함으로써 성능을 향상시키는 방법을 제안한다. 시뮬레이션 기반 실험 결과, 우리가 제안한 가속기가 기존 가속기보다 평균적으로 1.26 배의 성능 향상을 보였다.

## Ⅰ. 서 론

최근 몇 년 동안, 대규모 데이터 처리와 복잡한 계산 요구가 증가함에 따라 효율적인 행렬 연산의 중요성이 부각되고 있다. 특히, 희소 행렬 연산은 과학적 계산, 데이터 분석, 그리고 머신 러닝 분야에서 중요한 역할을 한다. 희소 행렬 이란 대부분의 요소가 0 으로 구성된 행렬을 말하며, 희소 행렬-벡터 곱셈 연산 (Sparse Matrix-Vector Multiplication, SpMV)은 메모리 성능과 계산 효율을 높이는 것이 중요하다. 희소 행렬 연산의 중요성은 다양한 응용 분야에서 그 가치를 입증하고 있다. 예를 들어, 자연어 처리, 이미지 인식, 그리고 추천 시스템과 같은 머신 러닝 알고리즘은 대규모 희소 데이터를 효과적으로 처리해야 한다. 대용량 데이터셋을 다루면서도 높은 정확도를 유지하려면, SpMV 연산의 가속화가 필수적이다. 한편, SpMV 는 희소 행렬에 대한 불규칙한 접근 패턴으로 인해 상당한 메모리 성능 저하가 일어난다 [1]. 자주 사용되고 중요하게 다뤄지는 희소 행렬에서 차원의 크기가 상당히 큰 경우가 많은데 이러한 경우에 성능 저하가 더욱 심해진다. 이런 문제를 해결하기 위해 [2]에서 Two-Step 이라는 알고리즘을 적용한 가속기를 제안하였다. 이 Two-Step 알고리즘을 적용하면, DRAM 에서 가져온 희소 행렬에 대해 여러 연산 유닛 (Processing Elements, PEs)를 사용하여 병렬 연산을 가능하게 함으로써, 상당한 성능 향상을 보일 수 있다. 그러나 DRAM 에서 가져오는 희소 데이터의 분포 특징에 따라 각 PE에 대한 부하 불균형 (load imbalance) 문제가 일어난다. 이는 PE 의 추가가 성능 향상에 기대만큼 기여하지 못하도록 한다.

따라서, 본 논문에서는 Two-Step 알고리즘에서 추가 PE 로 부하 불균형을 완화하는 방법론을 제안한다. 이를 위해, 과부하가 걸린 PE 의 부하를 분담해주는 PE 를 추가 배치하는 가속기 구조를 제안한다.

## Ⅱ. 본론

**배경설명 (Two-Step 알고리즘 기반 SpMV 가속기):** Two-Step 알고리즘이란 행렬 (A)와 벡터 (x)에 대해서 SpMV 연산을 할 때, 그림 1 과 같이 두 스텝으로 나누어서 연산을 하는 것이다. 첫번째 스텝에서는, 가로 및 세로로 균일한 크기로 나뉘어진 행렬 ($A^0_0$, $A^0_1$, … )과 가로로 균일한 크기로 나뉘어진 벡터 ($x^0_0$, $x^0_1$, …)에 대해서 행렬 곱셈 연산을 하여 중간 벡터 ($y^0_0$, $y^0_1$, …)를 도출하고, 두번째 스텝에서는 중간 벡터들을 병합하여 결과 벡터 z 를 도출하는 것이다. Two-step 알고리즘 가속기[2] 에서는 행렬 A 와 벡터 x 의 곱셈을 ($A^0$, $A^1$, …) 순으로 Step 1 작업을 한 후에, Step 2 단계로 중간 벡터 y 를 병합하여 결과 벡터 z 를 도출한다.

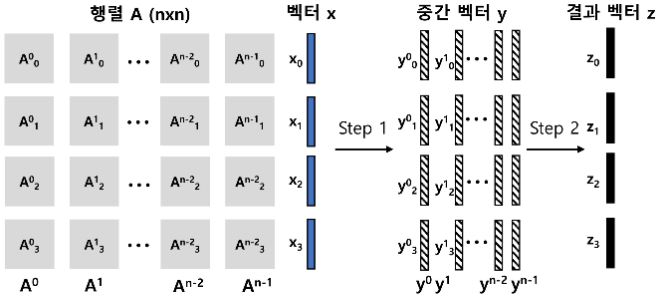


그림 1. Two-Step 알고리즘 도식도

**제안하는 아이디어:** 그림 2 는 본 논문에서 제안하는 Two-Step 알고리즘 기반 SpMV 가속기 구조를 묘사한다. 본 그림에서는 PE 가 4 개인 경우에 대한 예시를 보여준다. 만약 PE 가 4 개이면, 각 PE 는 4 개의 희소 행렬의 각 Row 의 데이터들에 대해서 곱셈 연산을

하게 된다. 희소 행렬의 분포도에 따라 각 Row 의 데이터 개수가 달라지게 되면, 부하 불균형이 일어난다. 그림 2 의 예시에서는 n 번째 Row 를 연산하는 PE 에 8 개의 입력 데이터가 할당되어 다른 PE 들에 비해 가장 많은 부하를 가지게 된다. 제안하는 가속기에서는 먼저 가장 많은 부하를 가지는 PE 를 찾고, 그 다음에 해당 PE 의 부하의 반 (그림 2 에서 빗금 표시된 블록들)을 가져와서 같이 연산을 하는 방식으로 진행한다. 그렇게 부하를 분배함으로써 읽어진 SpMV 의 성능을 가속화한다.

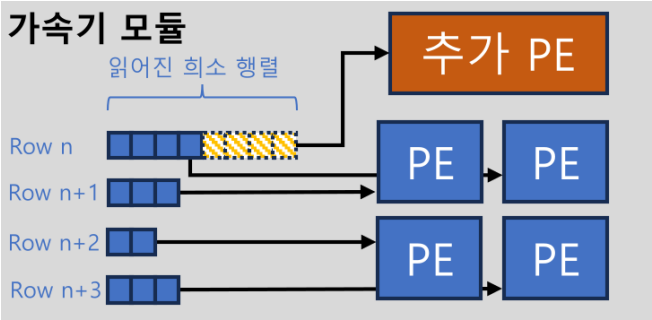


그림 2. 제안하는 Two-Step 알고리즘 기반 SpMV 가속기 구조

**실험 환경:** 본 논문에서는 DRAM 시뮬레이터인 Ramulator[4]를 변경하여 실험 환경을 구성하였다. 그리고, SuiteSparse[3]에서 가져온 데이터 (bcsstk32, webbase-1M, stomach, wiki-talk-temporal, 그리고 com-Youtube)에 대해서 기존 Two-Step 알고리즘 기반 SpMV 가속기와 제안하는 Two-Step 알고리즘 기반 SpMV 가속기를 평가하였다.

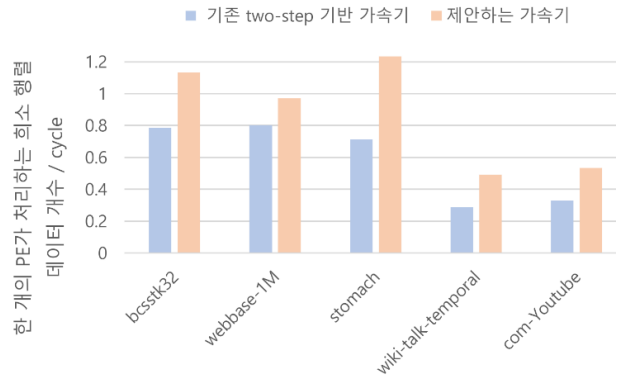


그림 3. 한 사이클당 한 개의 PE 가 처리하는 희소 행렬 데이터 개수 비교

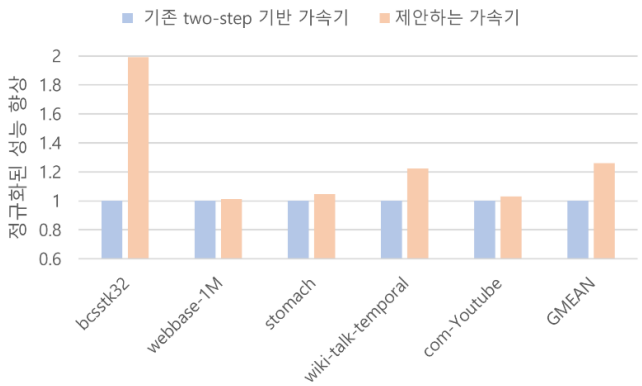


그림 4. Ramulator 기반 SpMV 가속 성능 평가 결과

**실험 결과:** 그림 3 은 한 사이클 당 한 개의 PE 가 희소 행렬에 대한 처리량을 보여준다. Stomach 행렬에 대한 처리량을 예시로 0.71 의 처리량에서 1.23 의 처리량까지 증가하였다. 평균적으로는 0.58 에서 0.87 까지 약 1.50 배로 처리량이 늘어난 것을 볼 수 있다. 한 개의 PE 가 추가됨으로써 전체적인 PE 효율이 증가함을 볼 수 있다. 그림 4 는 Ramulator 기반으로 기존 가속기와 우리가 제안한 가속기의 성능을 비교한 결과를 보여준다. 각 PE 의 데이터 처리 효율 향상을 통해 평균적으로 1.26 배의 성능 향상을 보였다.

## Ⅲ. 결론

본 논문에서는 대규모 데이터 처리와 복잡한 계산의 필요성이 증가함에 따라 희소 행렬 연산의 중요성과 그에 따른 성능 향상 방법을 탐구하였다. 희소 행렬의 불규칙한 접근 패턴으로 인한 성능 저하 문제를 해결하기 위한 이전 연구인 two-step 알고리즘 기반의 새로운 SpMV 가속기 구조에서 부하 불균형으로 인한 PE 효율 문제를 발견하였다. 본 논문에서 제안된 가속기는 부하 불균형 문제를 완화하기 위해 추가된 PE 가 다른 PE 의 부하를 분담하는 방식으로 설계되었다. 실험 결과, 제안된 가속기는 기존 가속기 대비 향상된 데이터 처리량을 보였으며, 이는 Ramulator 기반 실험을 통해 입증되었다. 특히, Stomach 행렬에서의 처리량 증가는 가속기의 성능 향상을 명확히 보여주는 예시로, 평균적으로 1.50 배의 처리량 증가를 관찰할 수 있었다


## ACKNOWLEDGMENT

본 논문은 2022 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2022-0-00153, 빔포밍 경로 분산을 이용한 AI 네트워크관리 기반 인빌딩용 O-RU 개발)


**추후 작성 예정**

## 참 고 문 헌


[1] Xie, Xinfeng, et al. "Spacea: Sparse matrix vector multiplication on processing-in-memory accelerator." *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021.

[2] Sadi, Fazle, et al. "Efficient SpMV operation for large and highly sparse matrices using scalable multi-way merge parallelization." *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. 2019.

[3] Timothy A Davis and Yifan Hu. The University of Florida sparse matrix collection. ACM Transactions on Mathematical Software (TOMS), 38(1):1-25, 2011.

[4] Kim, Yoongu, Weikun Yang, and Onur Mutlu. "Ramulator: A fast and extensible DRAM simulator." *IEEE Computer architecture letters* 15.1 (2015): 45-49.