2022-11-18

PIM-BLAS Framework

Embedded Systems and Computer Architecture Lab., Yonsei University

Enhyeok Jang <e@yonsei.ac.kr>
Chanyoung Yoo <chanyoung.yoo@yonsei.ac.kr>
Hongju.kal <hongju.kal@yonsei.ac.kr>

PIM-BLAS Framework

1. Overview

최근 산업계에서는 대규모 데이터를 활용하는 어플리케이션에 대한 전반적인 수요가 증가하고 있음. 하지만 이러한 어플리케이션에서 발생하는 빈번한 메모리 접근은 에너지 소비 증가 및 컴퓨터 시스템 전체의 성능 저하의 주요 원인이 되고 있음. 따라서 이러한 병목 현상을 해결하기 위해 메모리 내 처리(Processing-in-Memory, PIM) 기술이 학계 및 산업계에서 주목받고 있음.

한편 현재까지 개발된 PIM 구조에는 Processing-near-Memory(PNM), Digital-PIM(D-PIM), Analog-PIM(A-PIM) 등 다양한 형태가 존재함. 하지만 각 구조마다 동작 방식이 다르기 때문에 이들을 활용하기 위한 프로그래밍 방법 및 언어의 개발이 매우 난해한 상황임.

본 문서에서는 개발자들에게 기존에 제안된 다양한 종류의 PIM 구조에 적용할 수 있는 프로그래밍 방법을 제공하기 위해, 다종의 PIM 구조 및 프로그래밍 언어에 범용적으로 적용할 수 있는 프레임워크인 PIM-BLAS(Basic Linear Algebra Subroutines) 프레임워크에 대해 소개함. 이를 통해 프로그래머는 다양한 PIM 구조를 활용하는 딥 러닝 어플리케이션을 간편하게 구현할 수 있음.

2. PIM-BLAS Framework

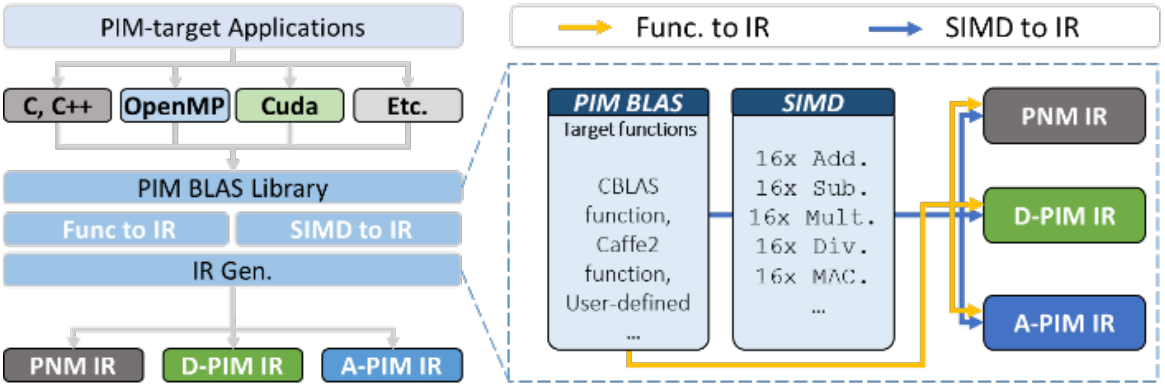
A. 개요

PIM-BLAS 는 프로그래머가 다종의 PIM 하드웨어를 활용할 수 있도록 API(Application Programming Interface) 형태로 제공되는 인공지능 및 빅데이터 향 프레임워크임. PIM 어플리케이션 개발자는 PIM-BLAS 에서 제공하는 CBLAS, Caffe2 및 다양한 사용자 정의 함수를 PIM 하드웨어에 적용 가능한 딥 러닝 어플리케이션 개발에 활용할 수 있음.

PIM-BLAS 는 프로그래밍 언어를 범용적으로 지원하기 위해 C, C++, OpenMP, CUDA 등에서 사용 가능한 공유 라이브러리를 제공함. 또한, 공유 라이브러리는 대표적인 3 가지의 PIM 구조인 PNM, D-PIM, A-PIM 을 모두 지원하기 위해 각각의 구조에 적합한 IR(Intermediate Representation)을 생성함.

B. IR 생성 기법

PIM-BLAS 는 두 가지 방식의 IR 변환 기법을 통해 사용자에게 다양한 개발 환경에 대한 편의성을



제공한다. figure 1 에서는 PIM-BLAS 에서 IR 을 생성하기 위한 두 가지의 기법에 대해 설명하고 있음.

Figure 1. IR 을 생성하기 위한 두 가지 기법

1. Func to IR 방식

Func to IR 방식은 주어진 PIM 하드웨어 제약 조건에 맞게 명령어의 개수를 최적화하여 고성능으로 실행할 수 있도록 BLAS 함수에서 IR 로 직접 변환하는 기법임.

2. SIMD to IR 방식

SIMD to IR 방식은 사용자에게 신규 API 에 대한 확장성을 제공하기 위해 BLAS 함수를 SIMD(Single Instruction Multiple Data) 연산으로 우선 변환한 후 이를 다시 IR 로 변환하는 방식임.

두 가지 IR 생성 기법에 따른 성능 비교를 위해 예제 파일을 이용하여 두 중간 언어의 수행 사이클을 비교하는 방식으로 성능 평가를 수행해 본 결과, func to IR 방식이 SIMD to IR 방식에 비해 32.2%의 성능 향상을 달성한 것을 확인함.

C. 다종의 PIM 구조 지원

	TensorDIMM	RecNMP	Newton / HBM	PRIME / ISSAC
Parallel Node	**DIMM-level**	**DIMM-level**	**Bank-level**	**X-bar-level**
Node Size	> 8 GB	> 8 GB	16 / 8 MB	32 KB
Data Size per Node	~GB scale	~GB scale	~MB scale	~KB scale
Data Parallelism	**Column-wise**	**Row-wise**	**Row-wise**	**Row-wise**

Figure 2. IR 을 생성하기 위한 두 가지 기법

본 프레임워크는 다양한 PIM 구조의 하드웨어적 특성을 고려하기 위해 각 구조 별 2 종의 PIM 모델을 선정하고[1][2][3][4][5][6], 사용하는 하드웨어 특성 및 관리 기법에 따라 병렬화 방식을 설계함. PIM-BLAS 에서는 Figure 2 와 같이 PIM 구조에 따른 병렬화 단계, 데이터의 크기 등을 파악하고 병렬화 방식을 결정함. 이를 통해 각각의 PIM 구조가 활용하는 병렬 연산 노드에 맞춰 데이터 배치 및 연산 할당이 가능함. 결론적으로, 사용자는 PIM-BLAS 를 활용하여 실행 중인 PIM 하드웨어의 사양과 구조에 관계없이 최적의 병렬 프로그래밍을 구현할 수 있음.

3. Conclusion

대규모 데이터를 활용하는 여러 딥 러닝 어플리케이션에서 발생하는 병목 현상에 대한 해결책으로 PIM 구조가 대두되고 있는 가운데, 개발자에게 여러 가지 PIM 구조에 적용할 수 있는 개발 환경을 제공하는 것은 매우 중요함. 본 보고서에서는 다종의 PIM 구조 및 프로그래밍 언어에 범용적으로 활용할 수 있는 프레임워크인 PIM-BLAS 를 소개하였음. PIM-BLAS 는 두 가지 IR 생성 기법과 병렬 구조 추상화를 통해 사용자로 하여금 다양한 환경에서 PIM 을 활용한 병렬 프로그래밍을 간편하게 구현할 수 있도록 할 것으로 기대함.

4. Reference

[1] Ke, Liu, et al. "Recnmp: Accelerating personalized recommendation with near-memory processing.", ISCA 2020.

[2] Kwon, Youngeun et al. "Tensordimm: A practical near-memory processing architecture for embeddings and tensor operations in deep learning." MICRO 2019.

[3] He, Mingxuan, et al. "Newton: A DRAM-maker's accelerator-in-memory (AiM) architecture for machine learning." MICRO 2020.

[4] Lee, Sukhan, et al. "Hardware architecture and software stack for PIM based on commercial DRAM technology: Industrial product." ISCA 2021.

[5] Shafiee, Ali, et al. "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars." ISCA 2016.

[6] Chi, Ping, et al. "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory." ISCA 2016.