암호학에서의 PIM 아키텍처 활용 동향 분석

서한결 [1], 이성우 [1], 이재용 [1], 황보현 [1], 장용근 [1], 오현영 [2*]
[1] 가천대학교 AI 소프트웨어학부 학부생
[2] 가천대학교 AI 소프트웨어학부 교수

{ha70801234, tigy99, sharon0320, ssogari0911, james4510, hyoh}@gachon.ac.kr

Survey on Cryptography with Processing-In-Memory

Han-Gyeol Seo, Sung-Woo Lee, Jae-Yong Lee, Bo-Hyeon Hwang, Yong-Geun Jang, Hyunyoung Oh
Dept. of AI · Software, Gachon University

요 약

대용량 데이터 처리 시 발생하는 CPU-메모리 간 성능 격차(performance gap)로 인한 병목 현상은 현대 컴퓨팅 시스템의 주요 과제이다. 본 논문에서는 이를 극복하기 위한 새로운 아키텍처인 PIM(Processing-In-Memory)을 소개하고, PQC(Post-Quantum Cryptography) 및 FHE(Fully Homomorphic Encryption)와 같은 암호학 분야에서 PIM 을 적용하여 처리 지연(latency)을 완화하고 성능 향상을 보인 사례를 분석한다. 각 구현 사례에 대해 기존 아키텍처 대비 구체적인 성능 향상 결과를 제시하며, 최신 연구 동향을 정량적으로 평가한다.

1. 서론

현대 컴퓨팅에서 '메모리 벽(Memory Wall)' 문제는 CPU 와 메모리 간 데이터 전송 지연으로 인해 성능 저하를 초래하는 주요 병목 현상이다. CPU 의 처리 속도는 Moore 의 법칙에 따라 18 개월마다 약 2 배씩 증가하고 있지만, 메모리 대역폭은 약 10 년마다 2 배 증가하는 수준에 그치고 있다[1]. 이로 인해 데이터 이동 과정에서 시스템 성능이 크게 저하되며, 특히 암호학 분야에서는 연산의 복잡성과 높은 자원 소모로 인해 기존 폰노이만 아키텍처만으로는 실시간 처리와 대규모 데이터 처리가 어렵다는 한계가 있다[2].

이를 해결할 방법으로 주목받는 기술이 바로 PIM(Processing-In-Memory)이다. PIM 은 데이터 처리를 CPU 가 아닌 메모리 내 또는 근처에서 수행하여, 데이터 이동에 소요되는 시간을 줄이고 성능을 최적화하는 혁신적인 아키텍처이다. 메모리 병목 문제를 완화하는 동시에 전력 소모를 줄일 수 있어 차세대 컴퓨팅 기술로 각광받고 있다. 본 논문에서는 특히 암호학 분야에서 PIM 을 접목한 사례들을 정리하고, 이를 통해 기존 아키텍처에 비해 성능 향상과 에너지 효율성을 정량적으로 분석한다.

2. PIM 아키텍쳐

PIM(Processing-In-Memory)은 크게 메모리 내에서 데이터 연산을 수행하는 PuM(Processing-using-Memory)과 메모리 근처에 프로세서를 배치하여 데이터 연산을 수행하는 PnM(Processing-near-Memory) 으로 구분된다. 단순히 하드웨어만을 사용하여 이러한 아키텍처를 구현할 경우, 오히려 성능 저하를 유발할 수 있다[3]. 이는 기존 메모리 셀의 밀도를 유지하면서 로직을 추가해야 하는 설계적 제약과, 메모리 내 연산의 특성을 고려한 최적화가 필요하기 때문이다. 따라서 최근에는 전용 연산 알고리즘을 병행하여 사용하는 방식이 연구되고 있으며, 이를 통해 성능 최적화를 도모하고 있다[4].

3. PIM 과 암호학의 통합

3.1 Post Quantum Cryptography

PQC(Post Quantum Cryptography)는 Shor 알고리즘을 필두로 한 양자 컴퓨팅의 등장으로 정수 인수분해 및 이산 로그 문제를 다항 시간 내에 해결할 수 있게 되면서, RSA(Rivest-Shamir-Adleman) 및 ECC(Elliptic Curve Cryptography)와 같은 기존 공개 키 암호 프로토콜이 취약해졌기에, 이들을 대체하기 위해 고안된 모든 기술들을 통칭한다.

PQC 는 구현 과정에서 매우 큰 차원의 행렬 계산과 모듈러 연산 등을 수행해야 하므로, 필연적으로 메모리와 프로세서 간 데이터 전송 과정에서 병목 현상이 발생하고, 이로 인해 처리 지연(Latency Overhead)이 발생하여 성능 저하를 유발하는 문제가

* 교신저자

있다.

이러한 처리 지연을 감소하기 위해, LBC(Lattice Based Cryptography) 구현 방식에 PIM 구조가 활용된 고처리량 NTT(Number Theoretic Transform) 기반 다항식 곱셈 아키텍처인 CryptoPIM 이 제안되었다. CryptoPIM 은 작은 다항식 차수(n ≤ 2^10)에 대해 뛰어난 처리량 향상을 보여주었으며, 에너지 효율성 또한 유지하는 성과를 보였다[5].

한편, Lin Ding et al.은 LPN(Learning Parity with Noise) 구현 방식에 PIM 가속기를 적용하였다[6]. 이를 통해 FPGA 및 CPU 프로세서를 활용한 기존 방식에 비해 처리 속도를 20.86 배에서 216.8 배까지 향상시켰으며, 기존의 PQC 표준(i.e., CRYSTALS-Kyber) 대비 15.4 배 적은 메모리 유닛을 사용하면서도 동일한 성능을 유지하는 결과를 얻었다.

암호 키를 생성하는 McEliece 암호체계 또한 대표적인 PQC 중 하나이다. McEliece 는 가우스 소거법 특유의 O(n^3) 시간 복잡도로 인한 느린 키 생성 속도가 문제로 지적되어 왔는데, C. Nugier et al.은 PIM 을 통해 키 생성 속도를 12.6 배, 암호화 속도를 최대 63 배까지 개선하였다[7].

3.2 Fully Homomorphic Encryption

FHE(Fully Homomorphic Encryption)는 데이터가 암호화된 상태에서도 연산을 수행할 수 있는 기술로, 데이터의 프라이버시를 유지하면서도 연산을 가능하게 하지만, 처리 속도와 자원 소모 측면에서는 매우 비효율적이다. 이러한 FHE 의 한계를 해결하기 위해 FHE-PIM 가속기가 고안되었다[8]. FHE-PIM 은 여러 클라이언트에서 암호문을 수신하고, 해당 암호문에 대해 연산을 수행하여 출력을 생성한다. 이를 위해 FHE-PIM 은 누적된 노이즈를 낮게 유지하여 결과 암호문이 올바른 클라이언트에 의해 해독될 수 있도록, PIM-Efficient Bootstrapping 기법을 사용한다. FHE-PIM 가속기를 활용한 FHE 의 Bootstrapping 및 산술 연산 성능을 측정한 결과, 기존 CPU 기반 시스템 대비 88,397 배의 예상 평균 처리량 향상을 보였다.

그러나, PIM 기반 FHE 가속기는 여전히 병목 현상을 유발하는 NTT 를 수행해야 하며, 이를 위해 복잡한 데이터 이동 특성을 지원하는 전문 하드웨어가 필요하다. 하지만, 이러한 하드웨어는 PIM 아키텍처를 복잡하고 비효율적으로 만들어 성능을 오히려 저하시키는 역효과를 유발할 수 있기에 이러한 복잡한 회로나 설계에 대한 의존성을 제거해주는 가속기의 개발이 요구된다.

4. 결론

PIM 아키텍처는 다양한 분야에서 메모리 병목 현상을 해결하기 위한 혁신적인 솔루션으로, 특히 대용량 데이터 처리가 필요한 작업에서 우수한 성능을 발휘한다. 암호학 연구에서는 PQC 와 FHE 와 같은 대용량 연산 작업이 필요한 암호화 기법의 구현에서 기존 아키텍처 대비 확연한 성능 향상을 확인할 수 있었다.

구체적으로, PQC 의 경우 데이터 전송 지연을 대폭 줄여 최대 216.8 배의 속도 향상을 달성했으며, FHE 의 경우 Bootstrapping 과 산술 연산의 성능을 88,397 배까지 향상시키는 등 기존 CPU 기반 아키텍처를 적용했을 때보다 월등히 빠른 처리 속도를 달성했다.

그러나, PIM 아키텍처의 도입에는 여전히 과제가 남아있다: 복잡한 회로 및 설계 의존성 감소, NTT 기반 아키텍처의 경우 높은 다항식 차수(n > 2^10)에서 에너지 효율 저하 문제, 메모리 셀 밀도와 로직 추가 사이의 트레이드오프 최적화가 그것이다. 이러한 한계점을 개선한다면, 더 높은 효율성을 갖춘 암호 프로토콜의 설계가 가능할 것이다. 향후 연구에서는 이러한 문제들을 해결하기 위한 새로운 PIM 아키텍처 설계 및 최적화 기법 개발이 필요할 것으로 보인다.

사사문구

이 논문은 2024 년도 정부(산업통상자원부)의 재원으로 한국산업기술기획평가원의 지원(No. RS-2024-00406121, 자동차보안취약점기반위협분석시스템개발(R&D))과 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. RS-2022-00166529)을 받고 과기정통부 정보통신기획평가원의 정보보호핵심원천기술개발사업(No. RS-2024-00337414)으로 수행한 결과임.

참고문헌

[1] A. Gholami, Z. Yao, S. Kim, C. Hooper, M. W. Mahoney and K. Keutzer, "AI and Memory Wall," in IEEE Micro, vol. 44, no. 3, p. 33-39, May-June 2024

[2] S. Gupta and T. Š. Rosing, "Invited: Accelerating Fully Homomorphic Encryption with Processing in Memory," 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2021

[3] O. Mutlu et al., “A Modern Primer on Processing in Memory,” arXiv:2012.03112 [cs.AR], 2021.

[4] F. Devaux, “The True Processing in Memory Accelerator,” in HCS, 2019.

[5] Nejatollahi, Hamid, et al, “CryptoPIM: In-memory Acceleration for Lattice-based Cryptographic Hardware”, 2020 57th ACM/IEEE DAC, San Francisco, CA, USA, July 2020, p1-6

[6] Lin Ding et al, “PIMA-LPN: Processing-in-memory Acceleration for Efficient LPN-based Post-Quantum Cryptography”, 2023 60th ACM/IEEE DAC, San Francisco, CA, USA, July 2023, page 1-6.

[7] C. Nugier and V. Migliore, "Acceleration of a Classic McEliece Post Quantum Cryptosystem With Cache Processing," in IEEE Micro, vol. 44, no. 1, pp. 59-68, Jan.-Feb. 2024, doi: 10.1109/MM.2023.3304425.

[8] S. Gupta and T. Š. Rosing, "Invited: Accelerating Fully Homomorphic Encryption with Processing in Memory," 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2021, page 1