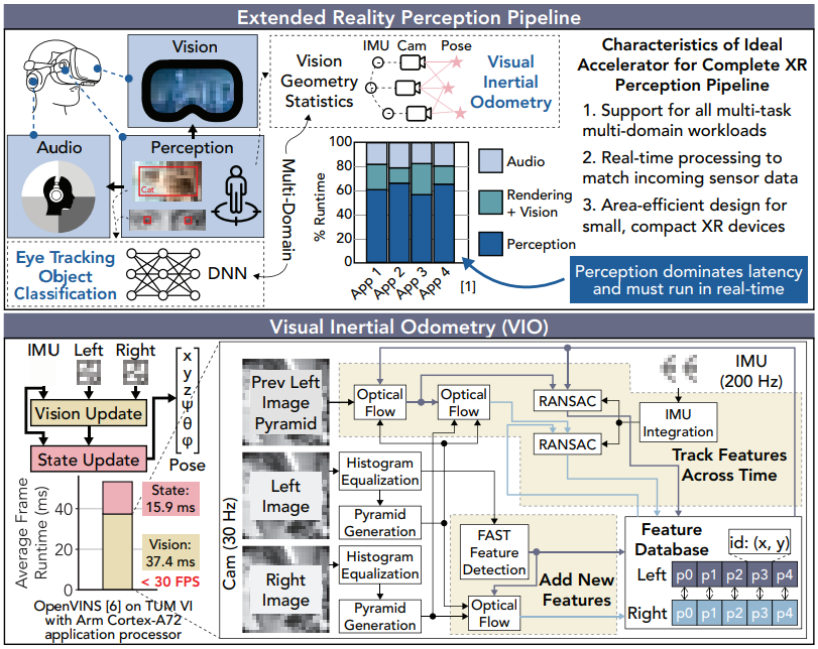
A-SSCC 2025 Review

경북대학교 전자전기공학부 박사과정 박승현

Session 3 Domain Specific Accelerators

이번 A-SSCC 2025 세션 3의 논문들은 서로 다른 응용 분야를 다루고 있음에도, 공통적으로 특정 도메인의 연산 패턴을 정확히 파악해 하드웨어 구조를 그에 맞게 최적화한 설계 접근이 돋보인다. 전반적으로 기존 기법을 실시간, 저전력 조건에 맞게 실제 칩으로 구현하는 데 초점을 맞춘 실용적 연구들이라는 인상을 준다.

#3-2 Birch: A Real-Time Accelerator for Multi-Task Mixed-Domain Extended Reality Perception Workloads



[그림 1] 실시간성이 중요한 XR에서 VIO와 DNN의 가속

이 논문은 extended reality (XR)에서 지각(perception) 파이프라인을 온전히 하드웨어로 끌어안으려는 시도를 보여주는 흥미로운 연구이다. Birch는 XR 시스템에서 공통적으로 발생하는 문제, 즉, visual inertial odometry (VIO)와 DNN 기반 인식 작업이 서로 다른 성격의 연산을 요구하면서도 동시에 실시간으로 처리되어야 한다는 점에 대해 하나의 SoC에서 균형 있게 대응하는 설계 결과이다. 기존 모바일 프로세서에서는 VIO의 vision update 단계가 상당한 연산량을 차지해 30 FPS 카메라 입력조차 안정적으로 처리하기 어려웠는데, Birch는 이 부분을 하드웨어 가속으로 분리함으로써 실시간성을 확보했다.

Vision accelerator의 구조적 특징은 비교적 명확한 편이다. FAST 기반 특징점 검출, 피라미드 생성, optical flow는 모두 반복적이고 데이터 접근 패턴이 뚜렷하여 하드웨어 파이프라인에 적합한 연산인데, Birch는 라인 버퍼, 정렬 버퍼, 더블 버퍼와 같은 고전적인 영상처리 기법을 각 단계에 적용해 CPU 대비 큰 지연 감소를 얻었다. 특히 Optical flow에서 더블 버퍼를 사용해 특징점 윈도우 계산과 위치 업데이트를 겹치는 방식은 구현 난도가 높지 않으면서도 latency 절감 효과가 꽤 크게 나타났다. 이러한 최적화 덕분에 vision update가 약 7ms 수준으로 줄었는데, 이는 단순히 '빠르다'는 의미보다는 카메라 프레임레이트를 안정적으로 수용하는 데 필요한 최소 조건을 충족했다는 점에서 현실적인 의미를 갖는다.

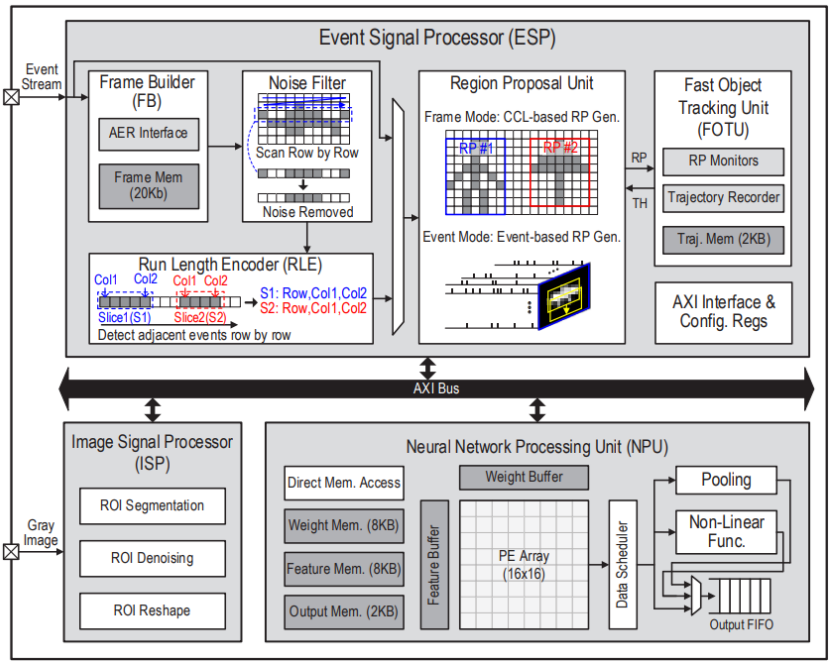
Birch가 DNN 연산 쪽에서도 실시간 성능을 확보한 점은 특이한 결과라기보다는, FP8 systolic array와 BF16 벡터 유닛을 함께 사용한 구조 덕분으로 보인다. XR에서 사용하는 DNN 모델의 규모는 대규모 모델에 비해 상대적으로 작고 고정된 패턴을 갖기 때문에, 32×32 systolic array 정도 규모의 매트릭스 유닛이라도 충분히 성능이 나온다. Eye-Gaze 모델이 1ms 미만으로 처리되는 결과나 ResNet-18이 10ms 이내에 처리되는 점은 이런 구조적 특징을 반영한 결과라고 할 수 있다. 다만 이는 Birch가 특별히 높은 연산 자원을 갖췄다기보다, XR perception에 필요한 모델들이 그리 크지 않기 때문에 현실적으로 가능한 수치로 이해할 수 있다.

흥미로운 부분은 BF16 벡터 유닛을 VIO의 RANSAC 계산에도 그대로 활용한다는 점이다. 이로 인해 별도의 전용 행렬 연산 블록 없이 CPU 대비 3배가량 빠르게 RANSAC을 수행할 수 있는데, 이는 면적 효율 측면에서 의미가 있다. XR과 같은 소형 디바이스 환경에서는 유닛을 공유하는 구조가 유리하며, Birch는 이런 선택을 적절히 적용한 사례로 보인다.

정확도 측면에서도 크게 무리 없는 수준이다. TUM VI와 EuRoC에서의 위치 추정 오차는 기존 VIO 연구 수준에 준하는데, 이는 Birch가 Vision Update만 가속하고 State Update는 CPU 기반 알고리즘을 그대로 유지하는 구조적 이유도 있다. 즉, Birch가 새로운 VIO 알고리즘을 제시한 것이 아니기 때문에 기존 소프트웨어 VIO의 정확도를 따라가는 결과가 나온 것으로 해석할 수 있다.

전체적으로 Birch는 XR perception 워크로드에서 가장 부담이 되는 Vision Update를 하드웨어로 옮기고, DNN 연산을 별도의 유닛으로 처리해 각 도메인에서 요구하는 성능을 실시간 수준으로 맞춘 일종의 균형 잡힌 설계로 평가할 수 있다. XR 기기에서 필요로 하는 조건을 무리 없이 충족한 실용적 접근을 보여주고 있다.

#3-3 A 96pJ/Frame/Pixel and 61pJ/Event Anti-UAV System with Hybrid Object Tracking Modes



[그림 2] 제안하는 anti-UAV 시스템의 하드웨어 아키텍처

이 논문은 이벤트 카메라 기반의 UAV 탐지 및 추적 시스템에서 흔히 나타나는 두 가지 문제—고속 이동 물체에서의 성능 저하와, 이벤트 기반 ODT (object detection & tracking)의 높은 전력—를 해결하기 위해, 프레임 기반과 이벤트 기반 방식을 상황에 따라 전환하는 하이브리드 아키텍처를 제안하고 있다. UAV처럼 작고 빠르게 움직이는 물체는 전통적인 프레임 기반 검출에서는 모션 블러가 문제이고, 반대로 이벤트 기반 방식은 항상 활성화되어 있어 전력 소모가 크다는 특성이 있기 때문에, 두 방식을 적절히 섞는 접근을 취한다.

전체 시스템은 ESP(Event Signal Processor), ISP(Image Signal Processor), NPU로 구성되며, 특히 ESP가 항상 켜진 상태에서 저전력으로 이벤트 스트림을 처리하고, 필요한 경우에만 NPU가 동작하도록 설계한 점이 눈에 띈다. 이 구조 덕분에 고정 전력 소모를 줄이면서도 빠른 Object Tracking이 가능해진다. ESP 내부에서 Run-Length Encoding을 사용해 이벤트 프레임을 slice 단위로 압축하고, Region Proposal Unit(RPU)이 이 slice들을 기반으로 객체를 찾고 추적하는데, 이 단계의 하드웨어가 논문의 핵심이라고 할 수 있다.

프레임 기반과 이벤트 기반을 전환하는 방식은 비교적 단순하지만 효과적이다. RPU는 먼저 프레임 모드에서 CCL 기반으로 객체 후보를 만든 뒤, 특정 크기 기준을 넘는 물체가 나타나면 이벤트 모드로 전환하여 주변 이벤트의 공간적 분포를 이용해 추적을 수행한다.

이 접근은 고속 이동 시 프레임 기반 검출의 불안정성을 보완하는 정도의 역할이며, 구현 복잡도 대비 성능 향상이 명확하다는 점에서 실용성이 있다. 실제로 이벤트 기반 RP 업데이트는 지정된 event threshold(TH)에 도달했을 때만 발생하게 하여 불필요한 업데이트를 줄이고, 객체 속도와 크기에 따라 TH를 조정해 RP가 과도하게 흔들리는 현상을 막는 식의 소규모 최적화가 포함되어 있다. 이런 방식은 특별히 새로운 알고리즘이라고 보긴 어렵지만, 하드웨어 구조에서는 오히려 이런 단순한 규칙 기반 방식이 안정적으로 동작할 수 있다는 장점이 있다.

고속 UAV에서 흔히 발생하는 문제는 RP가 실제 물체의 위치와 어긋나는 현상인데, 논문에서는 이를 FOTU(Fast Object Tracking Unit)로 대응한다. FOTU는 각 RP의 변화량을 모니터링하면서 TH 값을 유동적으로 조절하는데, 결과적으로 RP misalignment가 줄어들고 추적 품질이 안정된다는 실험 결과가 제시된다. 이런 방식은 기본적으로 heuristic 기반 조정이기 때문에 정교한 최적화라고 할 수는 없지만, 이벤트 기반 센서 데이터가 워낙 노이즈가 심하고 비균일한 특성을 갖기 때문에, 정교한 ML 모델보다 이러한 단순 조정이 더 안정적으로 동작할 수 있다는 점에서 타당한 접근으로 보인다.

NPU 부분은 전체 시스템에서 비중이 크지는 않지만, 전체 전력의 절반 이상이 이 유닛에서 발생한다는 점이 특징이다. 특히 NPU는 각 객체에 대해 한 번만 동작하도록 gating을 적용하여, 객체 수가 많지 않은 상황에서 상당한 전력 절감 효과를 얻는다. 전체적으로는 NPU를 매우 적극적으로 최적화하기보다는, 필요할 때만 켜는 구조를 통해 결과적으로 시스템 전력 효율을 높인 셈이다.

실험 결과로, 프레임 기반 모드에서 96pJ/(frame·pixel), 이벤트 모드에서 61pJ/event라는 수치는 저전력 Always-on 감지 시스템으로서는 합리적이며, UAV 탐지 정확도도 80% 후반대 수준으로 보고된다. UAV 속도가 높아질수록 이벤트 기반 trajectory 분류가 프레임 기반 패치 분류보다 안정적인 성능을 보인다는 결과도, 하이브리드 구조의 타당성을 지지하는 근거로 보인다.

종합적으로 보면 이 논문은 이벤트 기반 센서의 장단점을 하드웨어 레벨에서 자연스럽게 보완하는 구조적 설계 방법을 제시하였다. 프레임 기반과 이벤트 기반 중 어떤 방식이 더 우월한가를 논하기보다는, 두 방식을 상황에 따라 적절히 선택하는 것이 실제 시스템에서 더 안정적이고 전력 효율적인 접근이라는 점을 실험적으로 보여준 사례로 볼 수 있다. 이런 관점에서, UAV나 소형 고속 물체 감지 분야에서 하드웨어-알고리즘 공동 설계의 방향성을 확인할 수 있는 연구라고 정리할 수 있다.

저자정보

박승현 박사과정 대학원생

- 소속 : 경북대학교
- 연구분야 : 딥러닝 가속기 설계
- 이메일 : ijjh0435@gmail.com
- 홈페이지 : https://ai-soc.github.io/

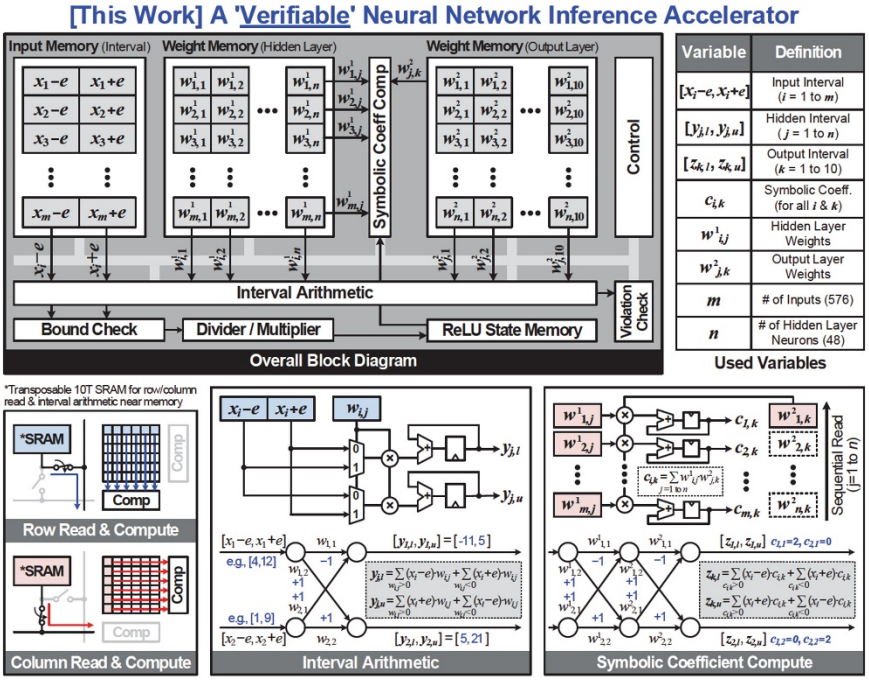
A-SSCC 2025 Review

연세대학교 전기전자공학과 석박통합과정 김동욱

Session 15 Techniques for Emerging Hardware Accelerators

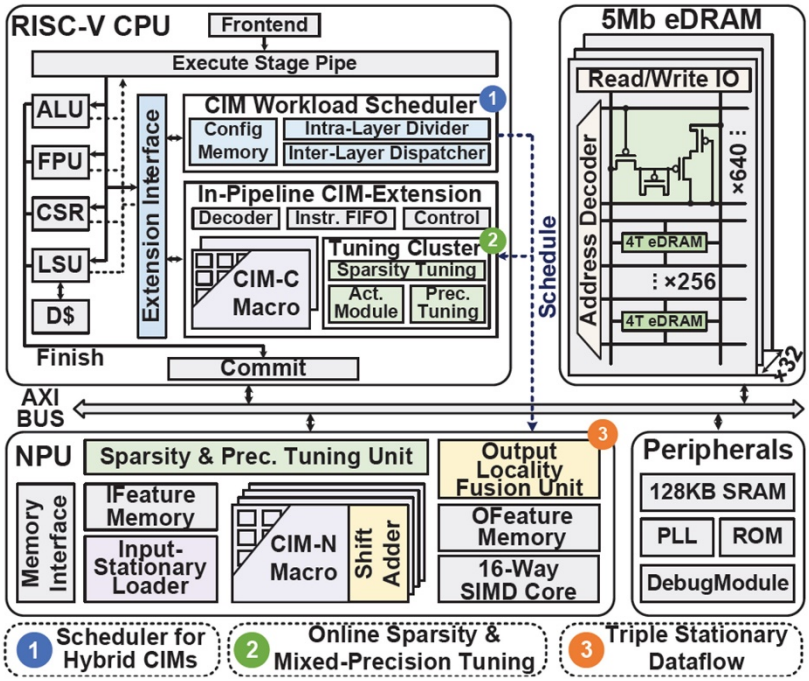
2025 A-SSCC의 Session 15는 Techniques for Emerging Hardware Accelerators라는 주제로 총 5편의 논문이 발표되었다. 이 세션은 edge AI 시스템의 궁극적인 성능 및 전력 효율 목표를 달성하기 위한 차세대 하드웨어 기술과 아키텍처에 중점을 두고 있다.

#15-1 KAIST와 University of California, Santa Barbara의 연구진이 발표한 본 논문 [Verifica: A 65nm In-Memory Symbolic Interval Computing Accelerator for a Formal Neural Network Verification]은 edge 디바이스를 위한, ASIC 기반의 formal neural network verification 하드웨어 가속기를 제안했다. 딥러닝 모델은 뛰어난 성능을 보이지만 노이즈에 의해 오류를 일으킬 수 있는 취약성을 가지는데, 이로 인해 edge 디바이스에서는 추론 가속기를 완전히 신뢰할 수 없다는 문제가 있었다. 본 연구에서는 이를 위해 딥러닝 모델의 신뢰성을 실시간으로 검증하는 하드웨어 측면의 해결 방안을 제시했다. 기존 소프트웨어 기반 검증 방식들은 많은 연산을 필요로 하고 독립적인 입력변수 판단으로 인해 느리고 부정확하다는 한계가 있다. 이를 보완하기 위해 symbolic interval analysis를 적용하지만, 본 연구는 소프트웨어 측면의 적용을 넘어 edge 디바이스 환경에 보다 적합하도록 이를 하드웨어로 최적화했다. 함께 적용된, 비선형함수인 ReLU의 선형 완화도 검증력 극대화에 기여했다. 또한 row와 column 연산이 모두 가능한 transposable 10T SRAM cell을 사용해서, 검증 연산에 대해 near-memory computation이 가능한 아키텍처로 구성했다. 본 연구의 이러한 알고리즘-하드웨어 co-optimization 결과, 65nm 테스트 칩에서 97.8% 이상의 검증 가능성과 1.22ms 미만의 검증 시간을 확인했다. 추가적으로, 에너지 소모 또한 검증 당 2.14μJ의 낮은 소비를 달성하여 edge 디바이스에 적용가능한 방법임을 입증했다.



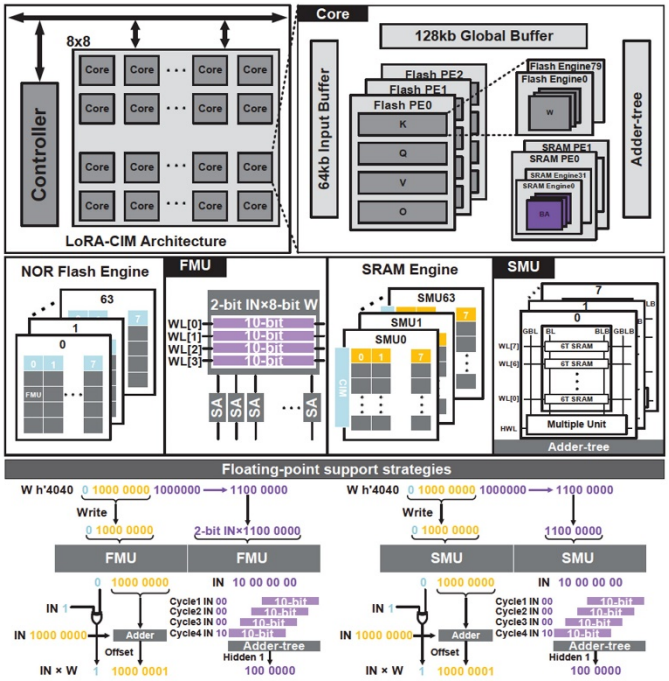
[그림 1] Verifica의 아키텍처와 회로 구성

#15-3 Peking University, Beijing과 Southeast University, Nanjing의 연구진이 발표한 본 논문 [A 30.28-151.42TFLOPS/W@FP8 Training Accelerator with Hybrid Compute-in-Memory and Workload Scheduling for Edge AI Applications]는 edge 디바이스에서의 모델 훈련에 최적화된 가속기를 제안했다. Edge 디바이스에서의 훈련에 대한 기존 연구들은 불균형한 연산 및 낮은 하드웨어 자원 활용률, 제한된 on-chip 메모리, 그리고 높은 메모리 접근 비용으로 인한 한계를 가지고 있다. 본 연구에서는 이러한 문제를 해결하기 위해 hybrid compute-in-memory (CIM) 기반의 아키텍처와 워크로드 스케줄링 기법을 적용한 훈련용 가속기를 제시했다. 두 개의 CIM macro인 CIM-C와 CIM-N을 각각 RISC-V CPU와 NPU 내부에 두어 hybrid CIM 아키텍처를 구성했고, CIM 워크로드 스케줄러는 RISC-V CPU 내에 위치하여 명령을 hybrid CIM에게 할당할지 여부를 판단하게 했다. 스케줄러는 워크로드 스위치 및 퓨전을 통해 유휴 상태의 CIM 코어에 작업을 할당함으로써 하드웨어 활용률을 높일 수 있게 했다. 또한, 훈련의 정확성과 효율성을 위해 동적인 sparsity 및 precision 튜닝을 구현했고, 기존 weight stationary 방식의 높은 입력 메모리 접근 비용 문제를 해결하기 위해 CIM-N은 input stationary와 output stationary도 지원하는 triple-stationary dataflow를 지원하도록 구현했다. 22nm 공정으로 제작한 칩 실험 결과, 기존 CIM 훈련 가속기 대비 2.9배 높은 성능을 기록했고 하이브리드 CIM 스케줄링을 통해 CIM 활용률을 평균 2.92배 개선했으며, 면적과 에너지 효율을 모두 고려한 FoM 지표에서 기존 CIM 가속기 대비 최대 28.08배 우수한 결과를 보였다.



[그림 2] 제안하는 hybrid CIM 기반 훈련용 가속기의 아키텍처

#15-4 Fudan University, China Flash Co.,Ltd, 그리고 Zhangjiang Laboratory의 연구진이 발표한 본 논문 [LoRA-CIM: A Fully-Digital Hybrid Flash-SRAM CIM Accelerator with LoRA Fine-Tuned LLM in Edge Devices]는 LLM을 edge 디바이스에서 효율적으로 fine-tuning 할 수 있도록 하는 hybrid CIM 가속기를 제안했다. LLM을 edge 디바이스에 배포할 때, 특정 작업에 맞게 fine-tuning하는 것은 필수적이지만 기존 CIM 기반 edge 디바이스는 작은 메모리 용량과 아날로그 CIM의 정확도 손실, 그리고 낮은 하드웨어 활용률 측면에서 문제가 있었다. 본 연구에서는 이러한 문제를 해결하기 위해 LoRA(Low-Rank Adaptation) 미세 조정 기법을 활용한 hybrid Flash-SRAM CIM 아키텍처를 제시했다. Flash CIM 엔진을 통해 용량 문제를 해결했고, 쓰기 작업이 없기 때문에 내구성을 확보할 수 있게 했다. 또한, 아날로그 CIM의 정확도 문제를 해결하기 위해 fully-digital 연산 아키텍처를 적용했다. 하드웨어 활용률 증가를 위해서는 연산 특성을 고려한 데이터플로우 최적화를 적용했는데, 트랜스포머의 레이어에 따라 CIM 코어를 파이프라인 모드로 운용하거나 병렬 모드로 운용하게 했다. 제작된 칩으로 실험을 진행한 결과, INT8 기준 1.45 TOPS/W, BF16 기준 700 GFLOPS/W의 효율을 달성했고, MobileLLM-125M 디코딩 속도를 SRAM 베이스라인 대비 7.7배 개선하였으며 에너지 감소 최대 43%를 달성했다. 본 연구를 통해 LoRA-CIM은 edge LLM 배포의 실현 가능성을 보였다.



[그림 3] Floating-point 연산 지원 방식과 함께 나타낸 LoRA-CIM 의 아키텍처

저자정보

김동욱 석박통합과정 대학원생

- 소속 : 연세대학교
- 연구분야 : 메모리 시스템
- 이메일 : dwkim3852@yonsei.ac.kr
- 홈페이지 : https://dtl.yonsei.ac.kr

A-SSCC 2025 Review

KAIST 인공지능반도체대학원 박사과정 하상우

Session 21 AI Accelerators

이번 IEEE ASSCC 2025의 Session 21은 AI 프로세서를 주제로 총 5편의 논문이 발표되었다. 올해 ASSCC Session 21에서 발표된 논문들은 총 두가지의 주요 트렌드를 보였다. 첫째, 거대 언어 모델 (LLM) 가속이 핵심 주제로 부상하였다 (논문 21.1, 21.4). 특히 단순히 엣지 디바이스에서의 single batch inference를 넘어, Adelia [1]처럼 여러 유저가 사용하는 multi-batch inference까지 고려한 논문이 등장하였다 (논문 21.1). 둘째, 확산 모델 (diffusion model)을 포함한 Embodied AI 가속의 중요성이 강조되었다 (논문 21.5). 마지막으로 엣지 디바이스를 위한 BNN 및 SNN 기반 초저전력 가속기들이 제안되었다 (논문 21.2, 21.3).

#21-1은 칭화대학교에서 발표한 28nm 멀티 칩렛 기반 LLM 가속기로, 멀티 유저 환경에서 LLM 서빙을 효율적으로 수행하기 위해 Prefill과 Decode 단계를 물리적으로 분리하는 분산 컴퓨팅(Disaggregated Computing) 방식을 채택한 것이 가장 큰 특징이다. Prefill 단계는 입력 토큰 전체를 한 번에 처리하는 행렬-행렬 곱셈으로 연산 집약적인 특성을 가지며, decode 단계는 토큰을 하나씩 순차적으로 생성하는 벡터-행렬 곱셈으로 메모리 대역폭 집약적인 특성을 가진다.

본 논문은 다중 배치 환경을 효율적으로 서비스하기 위해 칩렛 구조를 사용한 disaggregated computing이 필요하다고 주장한다. Disaggregated computing이란 prefill과 decode를 물리적으로 분리된 클러스터에서 독립적으로 처리하는 방식으로, 최근 서버단에서 LLM 서빙을 할 때 많이 사용되는 방식이다. 본 논문은 4개의 동일한 칩렛을 연결하고, 2개는 Prefill 클러스터 (P-C), 2개는 Decode 클러스터 (D-C)로 구성하여 각 단계를 독립적으로 최적화할 수 있게 하였다. 그림 1에서 볼 수 있듯이, disaggregated computing을 엣지 칩 클러스터에 적용하면 세 가지 문제가 발생한다. 본 논문은 이를 prefill 측면, decode 측면, 클러스터 간 통신 측면 총 세 가지로 나누어 모든 부분을 해결하고자 한다.

첫 번째 문제는 prefill 단계에서의 연산 병목이다. Prefill 단계에서는 데이터 재사용률 (OP/B)이 높기 때문에 가중치를 온칩 SRAM에 올려놓고 여러 토큰이 재사용하게 된다. 이로 인해 연산기는 최대 활용률로 계속 동작하게 되고, 이러는 동안 온칩 SRAM 대역폭

은 유휴 상태로 남게 된다. 이를 해결하고자 해당 논문에선 CFHM (Copy-less Float Hybrid-LUT Multiply)을 제안하였다. CFHM은 이 유휴 SRAM을 BF16 가수부 곱셈을 위한 참조 테이블 (LUT)로 활용한다. 핵심 아이디어는 하나의 SRAM을 16개의 뱅크로 분할하고, 64개의 입력 가수부를 하위 4비트 기준으로 16개 그룹으로 분류하여 각 뱅크에서 병렬로 테이블 참조를 수행하는 것이다. 이를 통해 prefill 처리량을 31.8% 향상시켰다.

두 번째 문제는 decode 단계의 메모리 병목이다. Decode 단계에서는 데이터 재사용률 (OP/B)이 낮기 때문에 외부 메모리에서 칩으로 계속 데이터를 가져와야 하고, 이 때문에 DRAM 대역폭이 병목이 된다. 반면 P-C의 DRAM 대역폭은 prefill 특성상 상대적으로 여유가 있다. 이 문제를 해결하기 위해 해당 논문은 HCWG (Hybrid Compressed Weight Gather)을 제안했다. HCWG는 prefill을 진행하는 클러스터에서 DRAM 대역폭이 남기 때문에, P-C에서도 가중치 데이터를 불러온 다음 이를 압축해서 D-C로 넘겨주는 방식을 사용한다. 압축 방식은 두 가지로 구성된다: (1) 허프만 지수부 압축 (HEC): 지수부가 가우시안 분포를 따르는 특성을 활용하여 상위 16개 빈도 값에 대해 단순화된 허프만 부호화 적용. (2) 활성화 기반 가수부 압축 (AMC): 입력 벡터의 부호 및 가수부 정보를 기반으로 내적 결과에 기여도가 낮은 가수부를 생략. 이를 통해 41.3%의 압축률을 달성하고 decode 단계에서의 메모리 접근으로 인한 지연 시간을 줄였다.

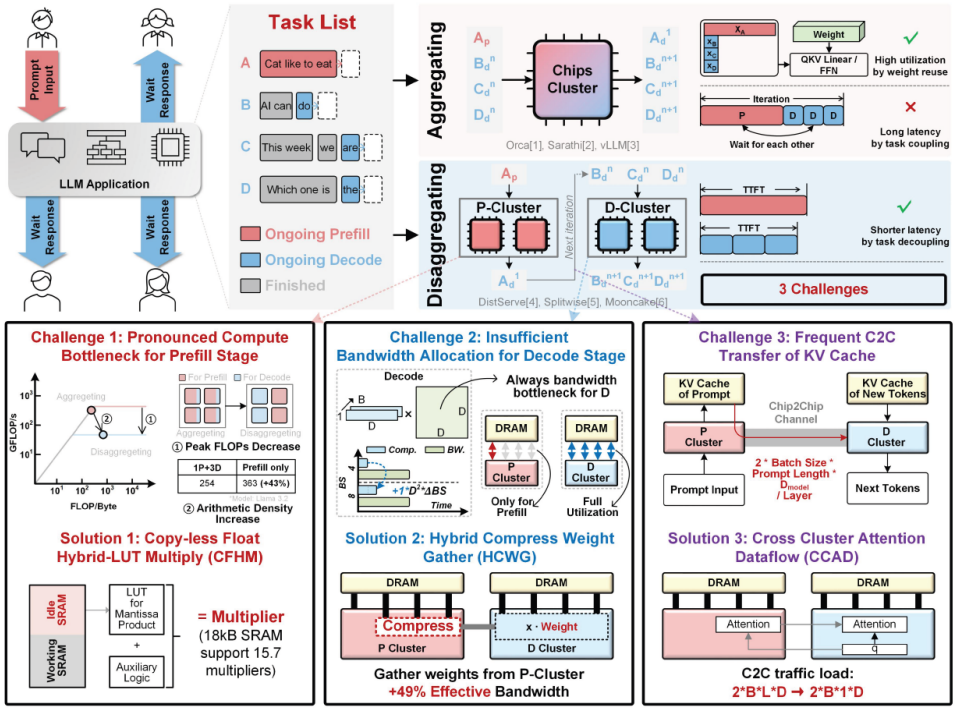
세 번째 문제는 KV 캐시의 칩 간 이동으로 인한 오버헤드이다. Disaggregated computing을 하면 P-C에서 생성된 KV 캐시를 D-C로 보내줘야 하기 때문에 에너지 및 지연 시간 오버헤드가 발생한다. 이를 해결하기 위해 본 논문은 먼저 UDB (Unified Dynamic KV Cache Balance)를 제안하였다. 어텐션 헤드 별로 KV를 P-C와 D-C에 나눠 저장하여 데이터 이동을 줄였다. 또한 CCAD (Cross Cluster Attention Dataflow)를 제안하여, P-C에 저장한 KV를 D-C로 가져와 연산하는 것이 아닌, 쿼리를 P-C로 전송하고 P-C에서 어텐션 연산을 수행하는 방식을 제시하였다. 이를 통해 칩 간 데이터 이동을 줄여 에너지 효율을 높이고 지연 시간을 46% 감소시켰다.

본 논문은 몇 가지 아쉬운 점이 있다. 첫 번째로, CFHM의 등장 배경이었던 Prefill단계의 문제점이다. 본 논문은 Prefill 단계에서 SRAM이 유휴 상태이기 때문에 비효율적이라고 주장하였다. 보통 엣지 디바이스에서 LLM을 돌리는 경우, decode 단계는 외부 메모리 대역폭이 병목이 되기 때문에 온칩 SRAM이 많이 필요하지 않다. 따라서 칩 설계 시 온칩 SRAM의 크기를 prefill 단계 요구량에 맞춰도 무방하다. 이러한 이유로 Prefill 단계에서 온칩 SRAM의 유휴가 문제가 된다는 주장은 설득력이 조금 부족하다.

두 번째 문제점은 HCWG로 인한 칩 간 데이터 이동 에너지 오버헤드가 명시되지 않았다

는 점이다. HCWG는 P-C에서 가중치를 읽어 압축 후 칩 간 링크를 통해 D-C로 전송하는 방식인데, 칩 간 데이터 이동 에너지는 일반적으로 상당히 크다. 따라서 압축으로 인한 DRAM 접근 속도 이득과 추가적인 칩 간 전송으로 인한 에너지 손실에 대한 정량적인 분석이 필요하다.

세 번째로 다양한 시나리오에 대한 검증이 부족하다. 다중 배치 환경에서는 프롬프트 길이와 생성 길이에 따라 prefill과 decode의 작업 부하 비율이 크게 달라진다. 그러나 본 연구는 다중 작업을 처리하는 경우 P-C와 D-C를 각각 2개씩 고정 할당해야 하기 때문에 이러한 작업 부하 변화에 유연하게 대응하기 어렵고 비효율적일 수 있다. 또한 CCAD와 UDB의 경우, 시나리오에 따라 사용하지 않는 것이 더 유리한 경우도 있을 것이다. 예를 들어 시퀀스 길이 L이 짧으면 KV 캐시 전송량 자체가 작아 쿼리 전송 방식의 이점이 감소하고, P-C가 prefill로 바쁜 상황에서는 어텐션 연산이 오히려 병목이 될 수 있다. 그러나 이러한 트레이드오프에 대한 분석이나 최적 지점이 어떤 경우인지에 대한 분석이 부족하다.



[그림 1] Disaggregated Computing의 문제점과 논문 21.1의 해결법

참고문헌

[1] J. -H. Kim et al., "Adelia: A 4nm LLM Accelerator with Streamlined Dataflow and Dual-Mode Parallelization for Efficient Generative AI Inference," 2025 Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)

저자정보

하상우 박사과정 대학원생

- 소속 : KAIST
- 연구분야 : 딥러닝 가속기 설계
- 이메일 : sangwoo_ha@kaist.ac.kr
- 홈페이지 : https://ssl.kaist.ar.k