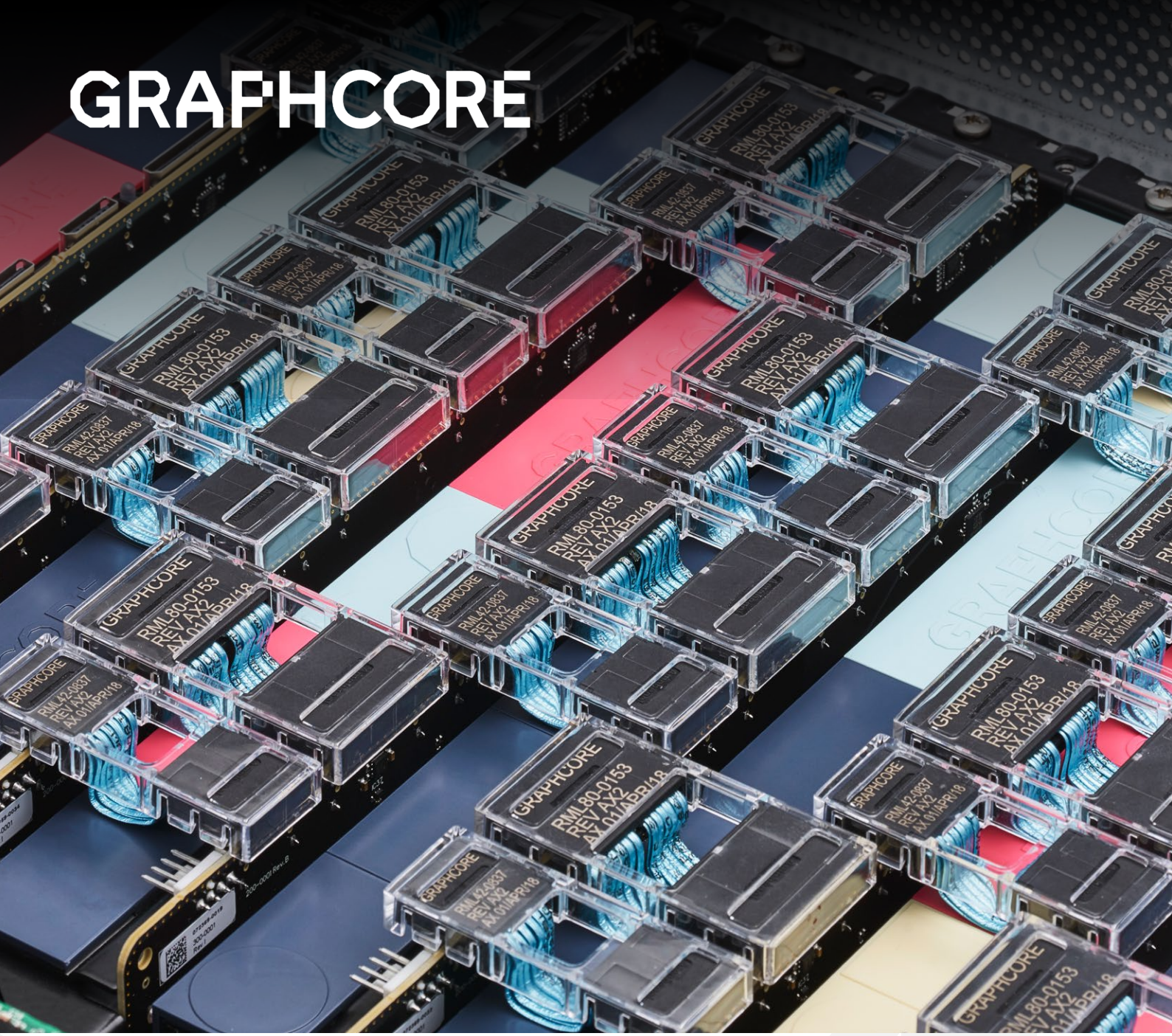


델 DSS8440 그래프코어 IPU 서버

백서

목차

IPU를 통한 머신 인텔리전스 컴퓨팅 구축 3

그래프코어 C2 IPU-프로세서 카드 6

델 DSS 8440 IPU-서버 7

포플러(Poplar®) SDK 9

Poplar SDK의 유연성 10

IPU-서버의 머신 인텔리전스 11

GCD(Graph Compile Domain) 기반 모델 병렬 처리 11

파이프라이닝(Pipelining) 12

재연산(Recomputation) 14

데이터 병렬 학습(Data Parallel Training) 또는 다중 복제 학습(Multiple Replica Training) 16

다수의 DSS 8440 IPU-서버 확장을 위한 인피니밴드(InfiniBand) 17

요약 18

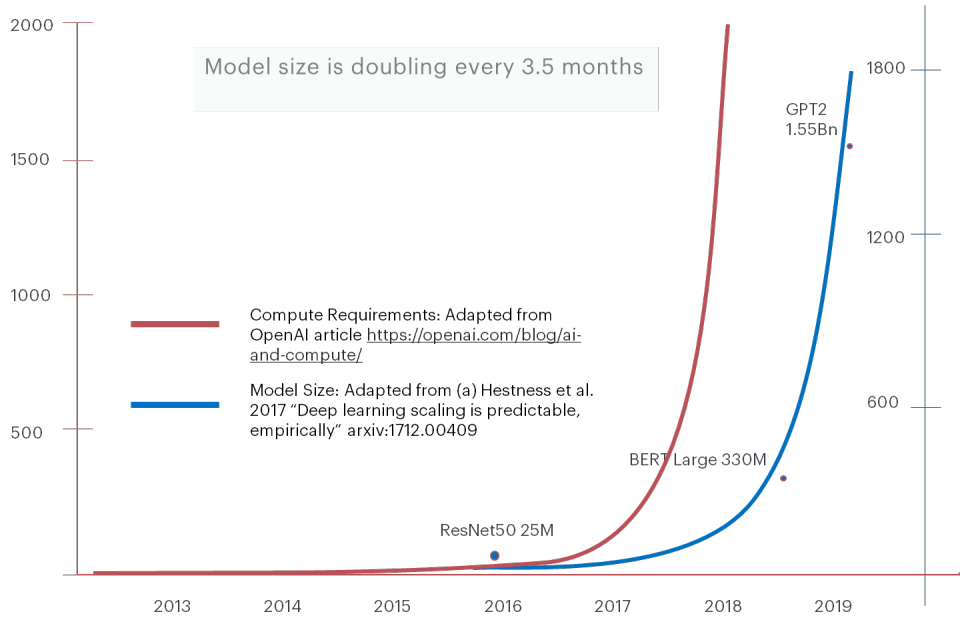
문의 사항 19

IPU 를 통한 머신 인텔리전스 컴퓨팅 구축

지난 70여 년간 인류는 컴퓨터 내 프로그램에서 단계별로 수행해야 하는 작업을 직접 지시해야 했습니다. 하지만 이제 컴퓨터는 머신 인텔리전스의 도입으로 데이터를 통해 학습하기 시작했습니다. 이는 마이크로프로세서와 전자 컴퓨터 자체의 발명 이후 컴퓨터 분야에서 나타난 가장 큰 변화입니다.

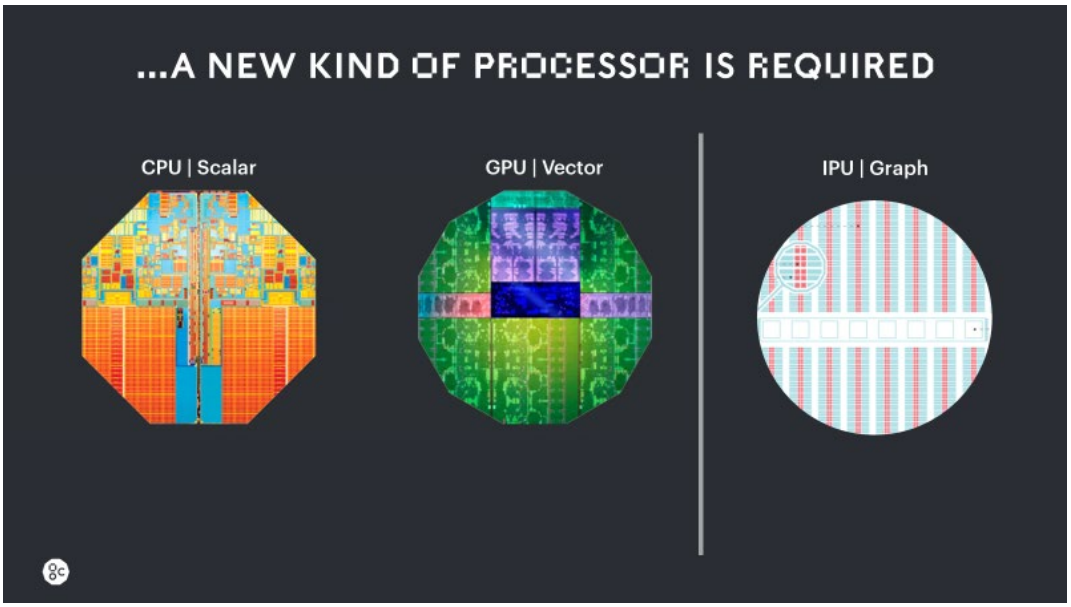
머신 인텔리전스 시스템은 불과 몇 년 만에 획기적인 발전을 이루었습니다. 첨단 머신 러닝 기술을 활용해 이미지를 정교하게 인식할 수 있을 뿐 아니라, 언어 구조를 이해하는 모델을 구축할 수 있습니다. 이처럼 다양한 시스템이 이미 상당한 영향을 받고 있지만, 이는 시작에 불과합니다. 보다 높은 수준의 인식 정확성과 효율성을 제공하고, 특정 작업에서 인간을 능가하는 머신 인텔리전스 시스템의 개발을 지원하는 새로운 유형의 머신 인텔리전스 모델이 구현되고 있습니다. 이를테면 버트(BERT: Bidirectional Encoder Representations from Transformers)와 같은 트랜스포머 기반 어텐션(Attention) 네트워크는 향후 풍부한 언어 이해를 위해 보다 효율적으로 수십억 또는 수조 개의 매개변수로 확장 가능한 희소 모델(sparse model) 구조를 활용하는 등 고효율적인 접근 방식이 필요해질 전망입니다. 이처럼 경험을 통한 학습이 가능한 머신 구축을 위한 기술 개발이 더욱 중요해지고 있습니다.

아래 그림은 일반적인 머신 인텔리전스 모델의 규모가 3.5개월 주기로 2배 가량 기하급수적으로 성장하고 있음을 나타냅니다. 또 컴퓨팅 요구 사항 역시 모델 규모의 2의 제곱에 비례하여 확장되므로 급속도로 증가합니다. 모델 규모의 성장에 따라 모델 정확성은 높아집니다. 머신 인텔리전스의 혁신은 매개변수를 줄이는 동시에 정확성을 높이는 데 중점을 두고 있으며, 이에 따라 분리 컨볼루션(separable convolution)이나 희소(sparse) 트랜스포머 등과 같은 기술이 요구됩니다.



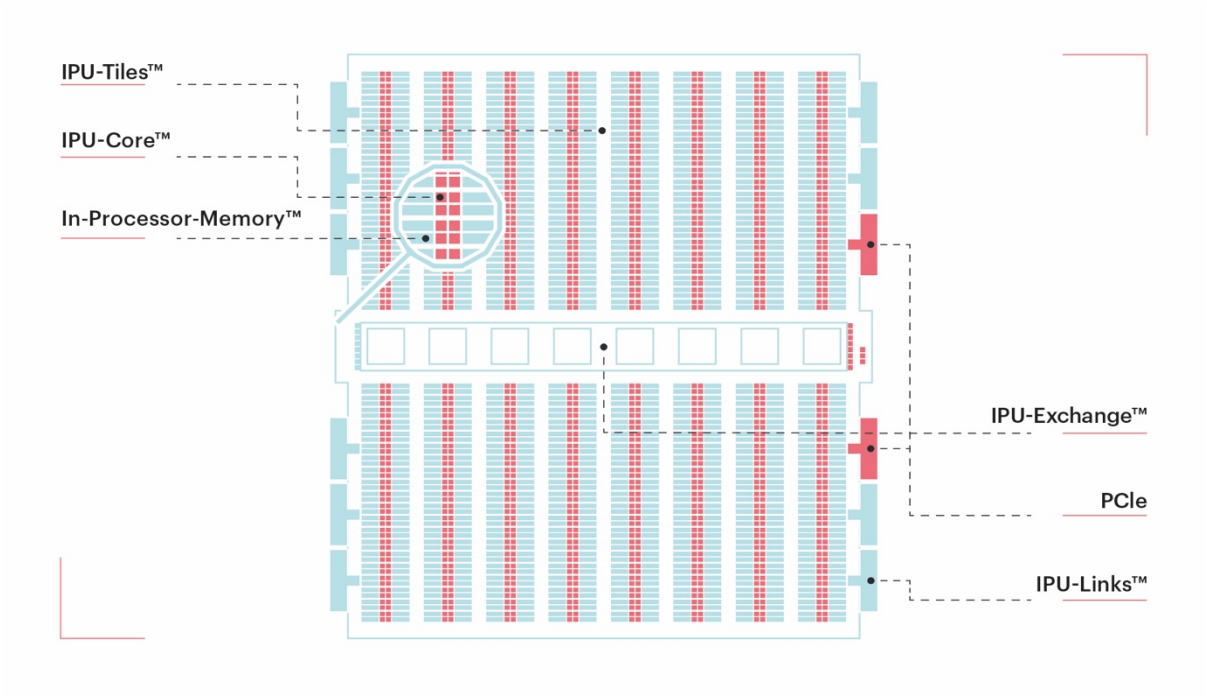
그래프코어는 머신 인텔리전스 분야의 많은 선도 혁신 업체들과 지속적인 협력 관계를 유지하고 있습니다. 해당 업체들은 기존 중앙처리장치(CPU) 및 그래픽처리장치(GPU)가 지속적인 혁신에 장애물로 작용한다고 밝혔습니다.

따라서 학습과 추론을 위해 더욱 복잡해진 지식 모델을 효율적으로 지원할 수 있는 새로운 유형의 프로세서가 필요합니다. 이것이 바로 그래프코어가 IPU(지능 처리 장치: Intelligence Processing Unit)를 개발한 이유입니다. IPU는 혁신 선도 기업의 요구 사항을 충족하고, 기업이 머신 인텔리전스 분야에서 새로운 돌파구를 모색할 수 있도록 돕기 위해 개발되었습니다.



IPU는 수천 개의 독립적인 프로세싱 스레드에서 대규모 병렬 프로세싱을 지원합니다. 이는 대용량 메모리 대역폭과 결합되어 수만 개의 병렬 스레드가 독자적으로 실행될 수 있도록 지원합니다.

그래프코어 콜로서스(Graphcore Colossus) GC2 IPU 아키텍처



새로운 유형의 머신 인텔리전스를 위해 설계된 그래프코어 IPU의 주요 특징은 다음과 같습니다.

- 수천 개의 독립적인 프로세싱 스레드에서 유연한 어드레스(address) 액세스를 위해 설계되었습니다.

- 메모리 대역폭 극대화 및 지연시간 최소화를 위해 프로세서 내 메모리(In-Processor-Memory)를 탑재해 프로세서 내부에 전체 지식 모델을 두고 있습니다.

- IPU-코어(IPU-Core)는 텐서(tensor) 작업에 최적화된 초고성능 혼합 정밀도 부동 소수점 연산을 지원하며, 난수 생성을 위한 하드웨어 지원을 포함하고 있습니다.

- 각 IPU-코어는 최대 6개의 독립 하드웨어 프로그램 스레드를 실행하므로 단일 IPU에서 지식 모델에 대해 병렬로 총 7천 개 이상의 개별 프로그램을 실행할 수 있습니다.

- IPU는 기존의 DNN 모델을 지원할 뿐 아니라, 확률론 학습 기반 시스템 및 혁신 전략 등 머신 인텔리전스의 혁신을 위한 새로운 방식을 제시합니다.

- IPU는 학습 및 추론에 대한 높은 효율성을 보이며, 머신 인텔리전스 컴퓨팅의 지연 시간을 최소화합니다.

각 IPU에는 1,216개의 고성능 머신 러닝 프로세서 코어(IPUCores)가 있으며 각 코어는 6개 프로세서 스레드를 포함하고 있습니다. 따라서, 연산이 실행되는 동안 정상 속도의 7천296개 프로세서 스레드가 병렬로 실행되며 각 IPU-코어는 100GFLOPS 이상의 컴퓨팅 성능을 발휘합니다.

각 IPU-코어는 256kB의 초고속 로컬 프로세서 내 메모리와 긴밀하게 결합되어 하나의 IPU-타일(IPU-Tile)을 형성합니다. 6개 프로세서 스레드는 직접 프로세서 내 메모리에 액세스할 수 있습니다. 전체 IPU는 약 300MB의 메모리와 45TBps의 대역폭을 보유하며 IPU가 메모리 대역폭 이슈를 우회하는 오프칩 메모리가 없습니다.

EEE FP32 및 FP16 데이터 포맷이 지원되며 IPU는 심층 신경망 내 모든 계층을 처리하고 250 테라플롭스 혼합 정밀도 16.32비트 연산 컴퓨팅을 실행할 수 있습니다.

각 코어는 62.5Tbps의 데이터를 전송할 수 있는 다이(die)의 중간에 있는 크로스바인 IPU-익스체인지(IPU-Exchange)에 직접 연결됩니다. IPU는 16-레인(lane) PCI 익스프레스(PCI Express) 인터페이스를 통해 호스트 프로세서로 연결됩니다. IPU에는 10개 고속 IPU-Link™ 포트가 있으며, 각 포트는 각 방향으로 160Gbps의 대역폭을 제공하며 8 레인이 있습니다. IPU-Link 포트를 통해 IPU는 호스트 프로세서 또는 호스트 메모리를 통하지 않고 직접 데이터를 교환할 수 있기 때문에, 최대 450GB/s의 IPU-Link IPU 간 대역폭을 통해 다수의 IPU가 효율적으로 작동할 수 있습니다. 또한, 형태 엔트로피 인젝션(configural entropy)을 위한 난수 발생을 위해 하드웨어 지원이 제공됩니다.

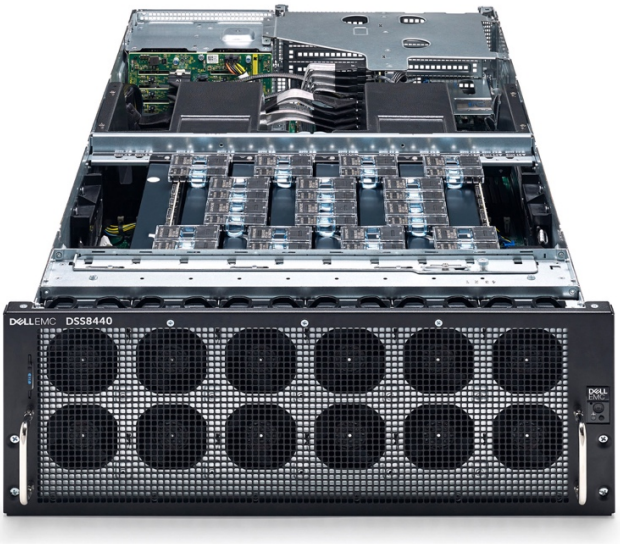
그래프코어 C2 IPU-프로세서 카드

그래프코어 C2 IPU-프로세서 PCIe 카드(Graphcore C2 IPU-Processor PCIe Card)는 2개의 IPU를 탑재한 듀얼 슬롯, 표준 높이 PCI 익스프레스 3/4세대 카드입니다. C2 카드는 각 링크(Link) 당 32Gbps 속도를 지원하여 총 2.5Tbps 또는 450GB/s 의 칩 간(chip to chip) 대역폭을 제공하는 80개의 IPU-Link 가 배치되어 있습니다. 2개의 IPU를 C2 카드 자체에 연결하기 위해 192GB/s의 IPU-Link 대역폭을 사용하며, C2 카드 간 연결을 위해서는 256GB/s의 IPU 대역폭을 사용해 다수의 C2 카드 내 IPU-프로세서가 연속 처리 리소스로 작동될 수 있도록 지원합니다(아래의 DSS 8440 시스템 토폴로지 그림 참조).

카드 내 두개의 IPU-프로세서로 2416 개 이상의 개별 프로세서 코어에서 총 200 TFLOPS의 연산 속도를 제공합니다. 이에 따라, 315W 카드는 DSS 8440 IPU-서버에서 수동 냉각됩니다.

델 DSS 8440 IPU-서버

DSS 8440 IPU-서버는 머신 인텔리전스 애플리케이션에 탁월한 고성능을 제공하도록 설계된 2소켓, 4U 서버로서, 총 1.6PetaFLOPs의 혼합 정밀도 머신 인텔리전스 연산 처리 능력을 제공하는 8개 그래프코어 듀얼-IPU C2 카드를 탑재하고 있습니다.



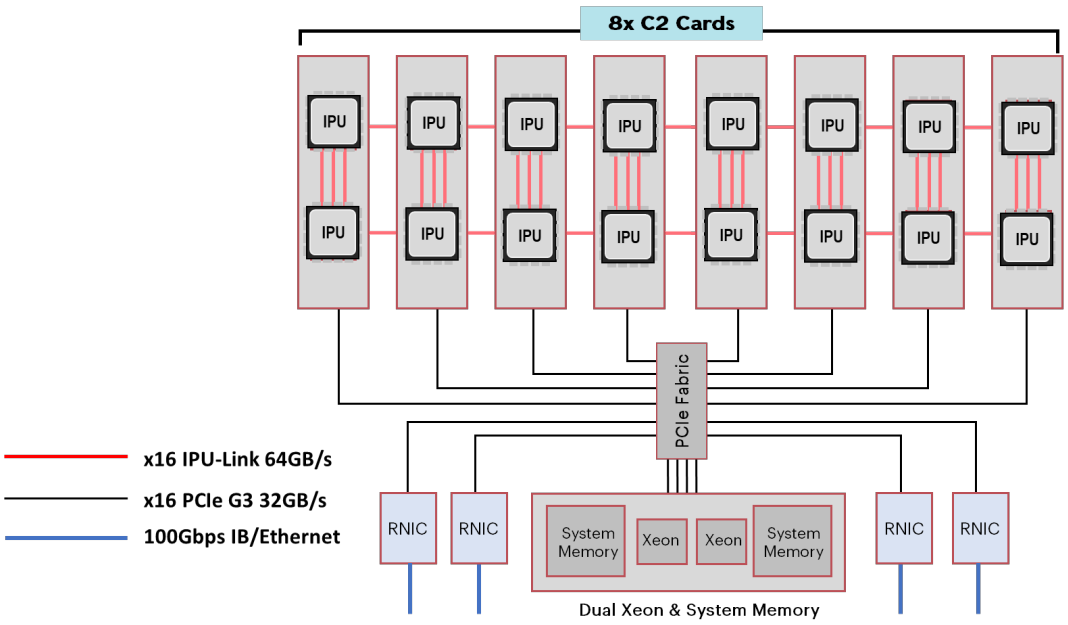
DSS 8440 IPU-서버는 C2 카드상의 2개 IPU 간에 192GB/s의 "단(rung)"으로 이루어진 IPU-Link "단계적(ladder)" 토폴로지와, C2 카드 간에 총 256GB/s의 양방향 대역폭을 활용합니다. 해당 배치를 통해 IPU 및 C2 카드 전반에서 모델, 매개변수, 활성화 및 데이터를 고속으로 공유할 수 있습니다.

C2 카드는 고속 PCIe Gen3 스위칭 패브릭을 통해 DSS 8440 IPU-서버 Xeon 호스트 CPU로 연결됩니다.

델 DSS8440 그래프코어 IPU 서버 백서(2020년 2월)

DSS 8440 IPU-서버 내에 4개의 개별 PCIe 스위치가 있지만, 이 경우에서는 “블랙박스(black box)” PCIe 패브릭으로 토폴로지 형태로 나타납니다. IPU는 독립적으로 자체 머신러닝(ML) 프로그램 실행이 가능하므로 호스트 CPU가 머신 인텔리전스 프로세싱 작업에 직접 관여하지 않습니다. 관련 설명은 포플러(Poplar) SDK 부분에서 보다 자세히 다룰 예정입니다.

나아가 DSS 8440 IPU-서버는 광범위한 로컬 스토리지를 탑재했으며, IPU는 해당 호스트 시스템 메모리를 직접 사용할 수 있어 대규모(수십억 개 매개변수) 모델을 수반하는 애플리케이션에 특히 유용합니다.



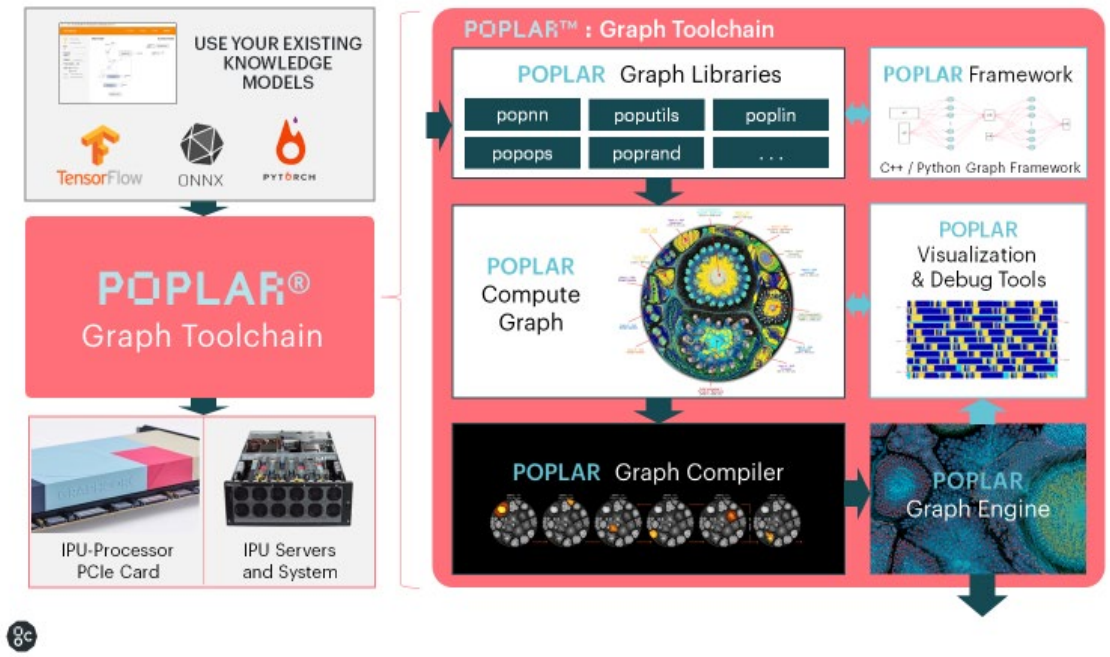
분산형 학습 애플리케이션의 경우, DSS 8440 IPU-서버는 일반적으로 기본 듀얼 100Gpbs 인피니밴드(InfiniBand) 어댑터를 이용해 서버 간 확장을 위한 다수의 100Gbps 네트워크 링크를 제공합니다. 해당 DSS 8440 IPU-서버 토폴리지 그림은 데이터 및 모델 병렬 구성에서 학습 또는 추론을 위한 단일 머신 인텔리전스 모델 구현을 검토할 때 사용됩니다.

이와 같은 기능적 특징을 바탕으로 DSS 8440 IPU-서버가 머신 인텔리전스 분야의 학습 및 추론 애플리케이션에 탁월하다는 점을 확인할 수 있습니다.

포플러(POPLAR®) SDK

머신 인텔리전스 애플리케이션은 그래프코어의 포플러(Poplar) SDK를 활용하는 DSS 8440 IPU-서버를 위해 개발되었으며, 오늘날 업계를 선도하는 주요 엣지(edge) 모델에서 즉시 사용 가능한 최첨단 성능을 제공합니다. 포플러 SDK는 텐서플로우(TensorFlow), 페이스북 파이토치(Pytorch), ONNX(Open Neural Network Exchange) 등 보편적인 AI 모델 생태계와 통합 연계 가능합니다. 포플러 SDK는 높은 수준의 머신 인텔리전스 그래프 기술을 수용하고, IPU에 적재되는 최적화된 포플러 그래프와 관련 컨트롤 프로그램을 컴파일합니다. 대용량 프로세서 내 메모리는 전체 모델이 IPU에 로딩되어 대용량 프로세서 내 메모리 대역폭을 적극 활용할 수 있음을 의미합니다. 또한 포플러 SDK는 일반적으로 수십 억 개 매개변수가 있는 차세대 자연어처리(NLP) 모델을 지원하는 등 모델 병렬 처리 및 파이프라이닝 접근 방식을 위한 다양한 지원을 제공합니다. 포플러 SDK의 주요 기능은 아래의 세부 설명을 참조하십시오.

포플러 SDK는 다양한 표준 프레임워크를 지원합니다. 텐서플로우와 연동된 경우, 직접적으로 XLA 그래프를 수용하고 XLA 출력을 포플러 그래프 및 컨트롤 프로그램으로 컴파일합니다. 그 외 머신러닝(ML) 프레임워크의 출력은 PopART™(Poplar Advanced Runtime)에 의해 중간 코드(intermediate representation)로 수용됩니다. 본질적으로 ML 프레임워크인 PopART는 이 외에도 다른 ML 프레임워크와의 긴밀한 통합을 지원합니다.



포플러 표준 라이브러리인 PopLibs는 커스터마이징이 가능한 다양한 연산자와 함수를 제공합니다. 소스 코드의 경우 고객들이 바로 사용하고 소스 개발에 기여할 수 있도록 제공되고 있으며, PopLibs는 2020년 내에 오픈소스로 전환되어 IPU 커뮤니티가 IPU와 관련한 혁신 기술들을 공개적으로 공유할 수

있게 될 전망입니다. PopLibs의 주요 특징은 다음과 같습니다.

• popnn - 신경망 함수들 (activation functions, pooling, loss)

• poplin - 행렬 곱셈(matrix multiply) 및 컨볼루션(convolution) 함수

• popops - pointwise 및 reduction 연산자

• poprand - 난수(random number) 함수

• poputil - 그래프 작성을 위한 유틸리티(uility) 함수

• popsolver – 모델 해석 함수

Poplar SDK의 유연성

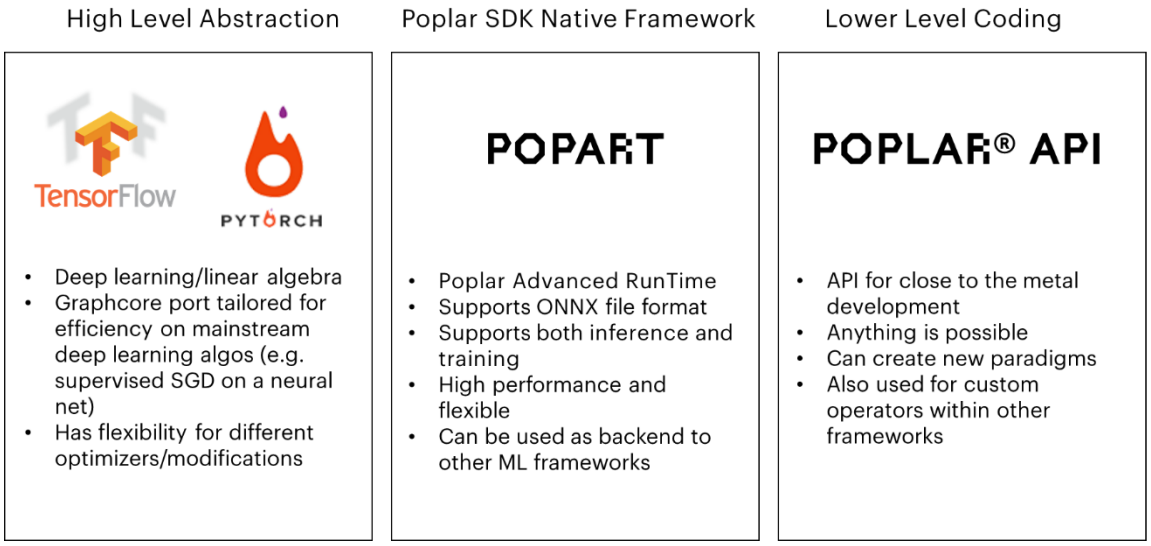
포플러(Poplar) SDK는 개발자의 배경과 목표에 따라 세가지 수준의 개발 방식을 제공합니다. 아래 그림은 포플러 SDK에 대한 3개 진입 지점을 설명합니다.

• 고수준 추상화(High Level Abstraction) – 텐서플로우(TensorFlow), 파이토치(Pytorch), ONNX 등 표준 ML 프레임 툴에서 설명된 기존 그래프를 통해 개발자들에게 쉬운 경로를 제공합니다. 포플러 SDK를 통해 변경 사항없이 기존 ML 프레임워크 모델을 구현하고 IPU에 최적화할 수 있습니다.

• PopART - Poplar Advanced RunTime은 추론 및 학습을 위한 높은 수준의 연산 그래프를

처리하는 포플러 SDK 네이티브 그래프 런타임입니다. PopART는 IPU별 그래프 최적화에 대한 지원은 물론, 개발자들이 대상 하드웨어에 더 가깝게 접근할 수 있도록 하는 기타 지원을 제공합니다.

• 저수준 코팅(Lower Level Coding) – 포플러 API는 개발자들이 IPU를 위해 직접 그래프를 코딩할 수 있도록 지원하는 직접적인 C++ 코딩 방식으로, 명시적으로 텐서(tensor), 점(vertices) 및 엣지 단을 지정하고 완벽하게 통제합니다. 이는 커스텀 연산자 또는 커스텀 네트워크 계층을 만들거나 기타 최적화 용도로 사용될 수 있습니다. 개발자들은 다양한 사례를 제공하는 PopLibs 소스코드에 액세스하여 개발 속도에 박차를 가할 수 있습니다.



IPU-서버의 머신 인텔리전스

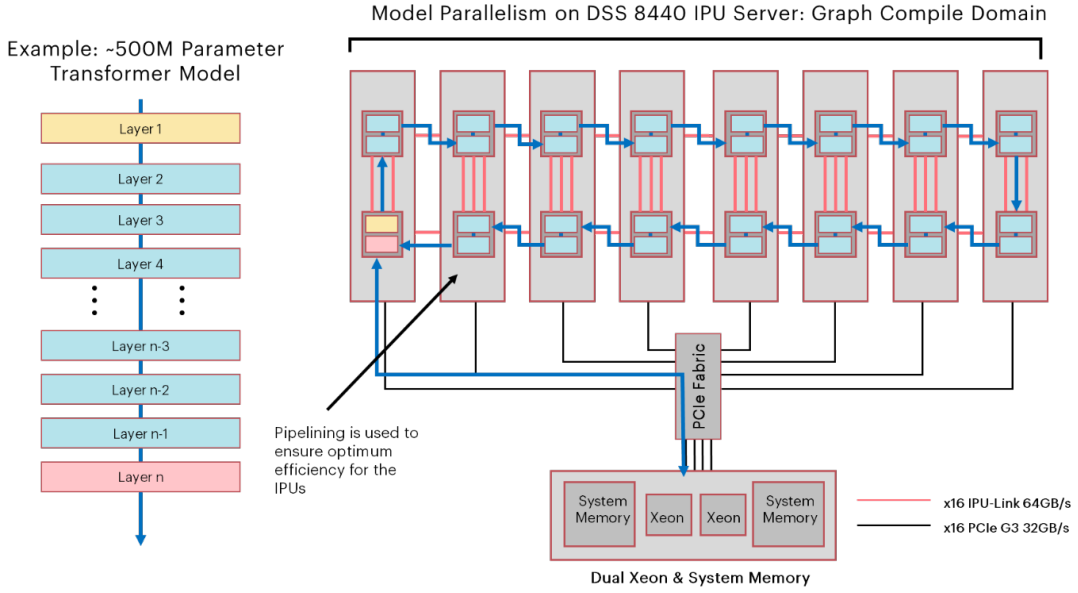
DSS 8440 IPU-서버는 기존 데이터 병렬 아키텍처는 물론, 효율적인 파이프라이닝 모델 병렬 아키텍처를 포함해 학습 및 추론을 위한 머신 인텔리전스 지식 모델을 구현하는 새로운 가능성을 제공합니다. 물론 최적의 구현 방식은 모델에 따라 달라질 수 있습니다. 하지만 IPU와 포플러 SDK는 머신 인텔리전스

프로세싱을 위한 차세대 플랫폼을 제공한다는 점에서 사용자에게 보다 효과적인 모델 성능 최적화를 구현할 수 있는 절호의 기회이기도 합니다. 현재 반도체 업계에서는 기존의 프로세싱 아키텍처가 일정 부분 자리잡은 상태이지만, 동시에 머신 인텔리전스에 대한 새로운 돌파구를 적극적으로 모색하고 있는 추세입니다. 차세대 IPU의 관점에서 모델 아키텍처를 파악한다면 혁신에 더욱 가까이 다가갈 수 있습니다.

IPU 아키텍처의 강점은 포플러 SDK의 혁신적인 기능을 통해 최적으로 활용될 수 있습니다.

GCD (Graph Compile Domain) 기반 모델 병렬 처리

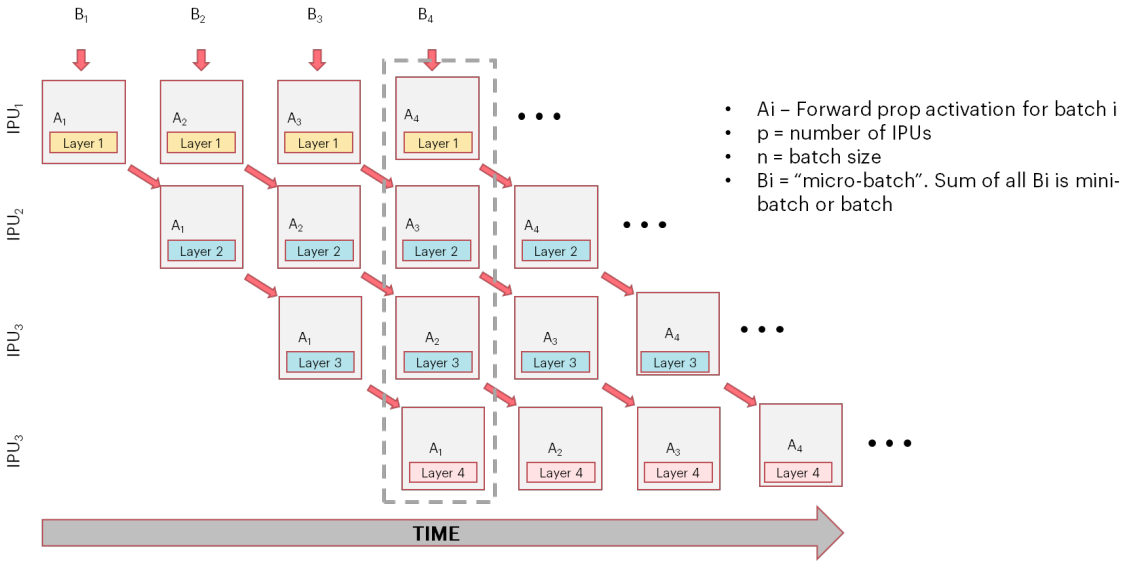
Graph Compile Domain(GCD)은 포플러 SDK가 모델 병렬 구현을 컴파일할 수 있는 IPU의 개수를 정의함으로써 대형 모델이 고대역폭 IPU 프로세서 내 메모리를 적극 활용할 수 있도록 지원합니다. GCD는 DSS 8440 IPU-서버 내 16개 IPU를 포함하며, 서버 내 개개의 IPU 전반에 모델을 '쪼개는(sharding)' 방식으로 대형 모델 분할이 가능합니다. 중소형 모델은 단일 IPU의 프로세서 내 메모리에 최적화 되어있으며, 전체 16개 IPU 전반에서 데이터 병렬 프로세싱을 실행할 수 있습니다.



특히 GCD는 파이프라이닝(아래 참조)과 결합될 경우, 초대형 그래프에서 지연 시간을 최소화하고 연산을 극대화하는 데 유용합니다. 위 그림은 대형 트랜스포머 기반 모델(일반적인 NLP 활용 사례)로, DSS 8440 IPU-서버 내 전체 16개 IPU에서 분할된 모델을 나타냅니다.

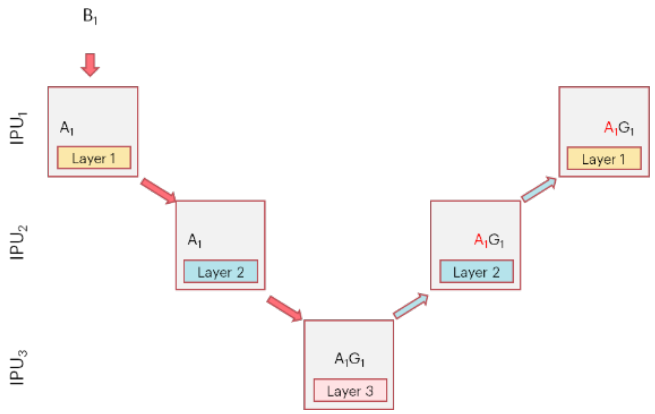
파이프라이닝(Pipelining)

파이프라이닝은 GCD를 통한 모델 병렬 구현을 결합했을 경우 모델 병렬 처리에 관련된 모든 IPU가 최대치로 활용될 수 있도록 보장하는 접근 방식입니다. 그 결과, 프로세서 효율성이 높아지고 처리량(throughput) 및 지연시간 기준에서 성능이 향상됩니다. 아래 그림은 추론을 위한 모델 병렬 접근방식에서 파이프라이닝을 사용하는 것을 집중적으로 보여주고 있습니다(점선으로 표시된 박스는 모든 IPU가 최대한 활용되는 파이프라인의 중심부입니다). 파이프라인은 모든 깊이(DSS 8440 IPU-서버 내 최대 16개 IPU)가 될 수 있습니다. 모델 병렬 시리즈 내 첫번째 IPU(IPU1)는 활성화(activation) 함수가 다음 IPU로 전달되면, 첫번째 배치(batch)(B1)를 수용합니다. IPU1는 계속해서 새로운 배치(batch)를 수용하고 해당 배치(batch)에 대한 활성화 함수를 차례대로 전달합니다. 두번째 IPU가 활성화 함수를 수용하고, 다음 레이어를 처리하면, 그 아래로 내려갑니다. 파이프라이닝은 모든 IPU를 활성화시키고 배치(batch) 4 (B4)에 의해 모든 IPU는 능동적으로 병렬 처리를 실행합니다. 해당 프로세싱 파이프라인은 추론 상으로 비교적 단순한 반면, 학습 상에서 역전파(back propagation)로 인해 보다 복잡해집니다.



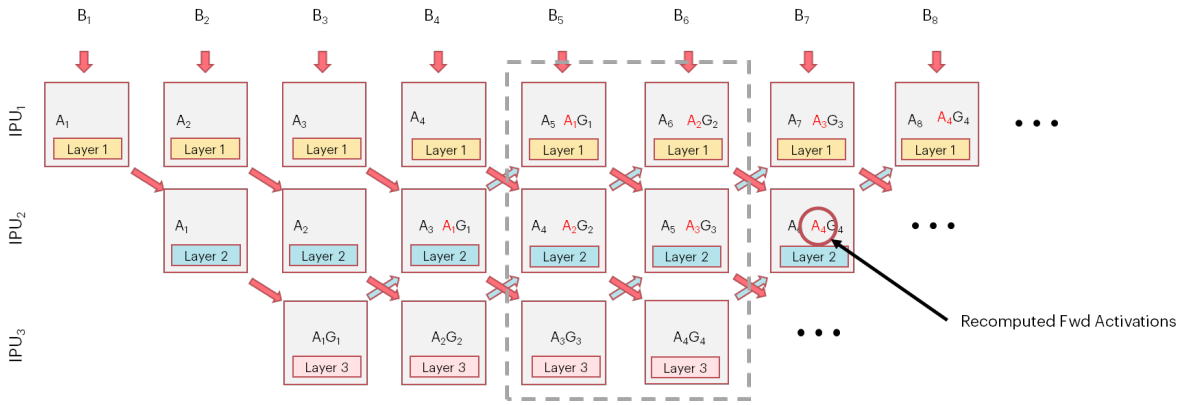
학습을 위해, 파이프라이닝은 정방향 계산(forward pass)은 물론, 역전파(back propagation) 및 가중치 업데이트(weight update) 모두를 수용해야 합니다. 아래 그림은 단일 정방향 계산(forward pass)과 역전파(back propagation)를 보여준 다음, 중첩 배치(overlaid batch)가 있는 전체 파이프라인을 병렬로 나타냈습니다. 점선으로 표시된 박스는 관련 IPU들이 최대한 활용되는 파이프라인의 중심부(원하는 수준의 깊이로 설정할 수 있으며, 깊이가 클수록 배치(batch) 사이즈가 증가함)를 나타낸 것으로 정방향 계산(forward pass) 활성화를 처리하며 저장된 활성화 입력(input)에서 이전 활성화를

재계산하고 그래디언트 업데이트(gradient update)를 계산합니다. 재계산(아래 참조)은 개발자들이 고대역 프로세서 내 메모리를 최대한 활용할 수 있도록 한다는 점에서 이 접근방식을 위한 핵심 요소입니다.



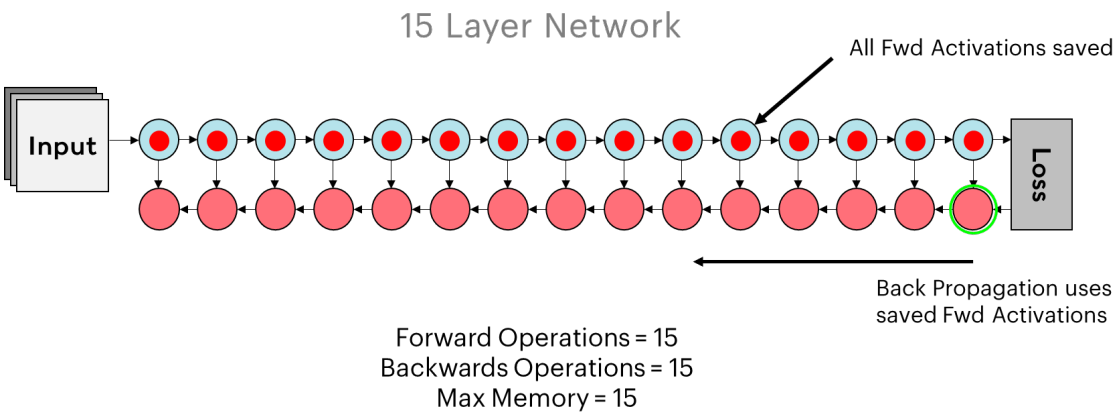
단일 정방향 계산(forward pass) 및 역전파(back propagation)와 다수의 IPU 전반의 병렬 모델(최대 GCD=16).

IPU 활용도를 극대화하기 위해 다수의 정방향 계산(forward pass)과 역전파(back propagation)를 중첩한 파이프라인 구조의 학습. 파이프라인의 중심부(점선으로 표시된 박스)에서 모든 IPU들은 정방향 계산(forward pass) 활성화를 능동 처리하고, 저장된 활성화 입력(input)에서 이전 활성화를 재계산하며, 그래디언트(gradient)를 계산한다는 점을 유념하십시오.



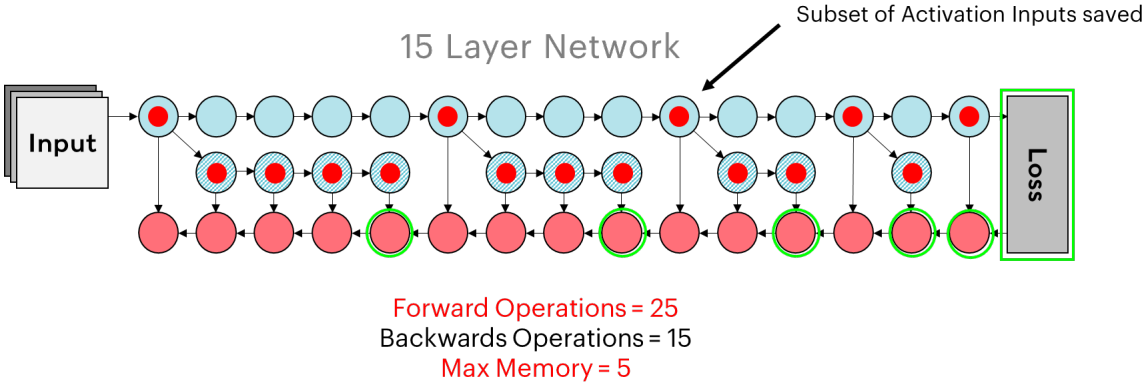
재연산 (Recomputation)

IPU 아키텍처의 주요 강점은 고대역폭/지연 시간 최소화를 지원하는 프로세서 내 메모리입니다. 재연산은 포플러 SDK의 기능 중 하나로, IPU 메모리 리소스와 IPU 고성능 연산 기능을 극대화합니다. 멀티 계층(layer) 모델에서 활성화는 계층 간에 계산되며 일반적으로 중간 결과값으로 저장됩니다. 포플러 SDK는 재연산을 통해 특정 활성화 입력(input)을 선택하고 이를 사용해 역방향 계산(backward pass) 시 요구되는 활성화에 대한 계산을 재실행합니다. 아래 첫번째 그림은 모든 정방향(forward) 활성화가 저장되어 역방향 계산(backward pass)에서 가중치 업데이트 계산을 지원하는 일반적인 접근방식을 제시합니다.



포플러 SDK는 재연산 시 메모리 절약과 TFLOP 지출을 최적화하는 우수한 활성화 입력(input)을

저장해 귀중한 프로세서 내 메모리를 보다 효율적으로 사용할 수 있도록 지원합니다. 아래 그림은 가중치 업데이트의 역방향 계산(backward pass)을 위해 모든 필수 활성화 히스토리를 재계산하는 데 저장된 활성화 입력(input)의 하위 집합이 사용되는 방식을 나타냅니다.



재연산은 IPU 아키텍처의 강점 외에도 다양한 이점을 제공합니다:

- ResNets 및 DenseNets의 경우 재연산은 NN을 학습시키는 데 필요한 메모리를 최대 5배까지 절감할 수 있으며 밀결합(tightly coupled) IPU 프로세서 내 메모리의 사용을 최적화할 수 있습니다.

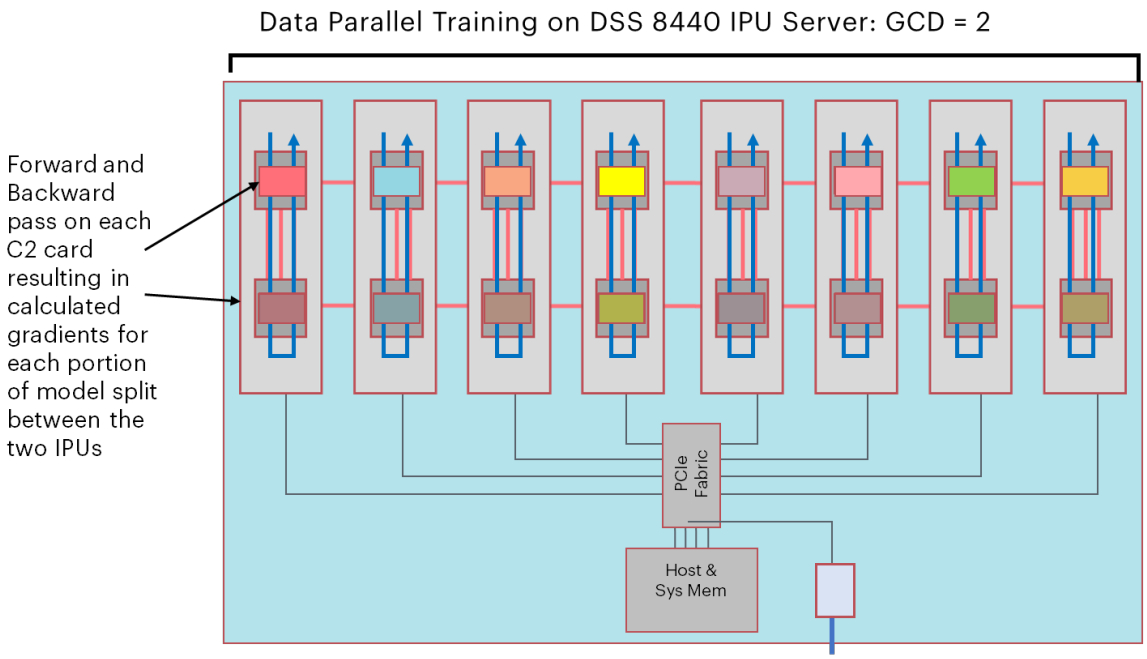
- 포플러 SDK는 대상 메모리 오버헤드에 대한 FLOP을 최소화하기 위해 제한된 메소드를 지원합니다.

- 추가적인 FLOP 컴퓨팅에 필요한 시간(wall clock time)은 대형 배치(batch)를 사용함으로써 상쇄될 수 있습니다.

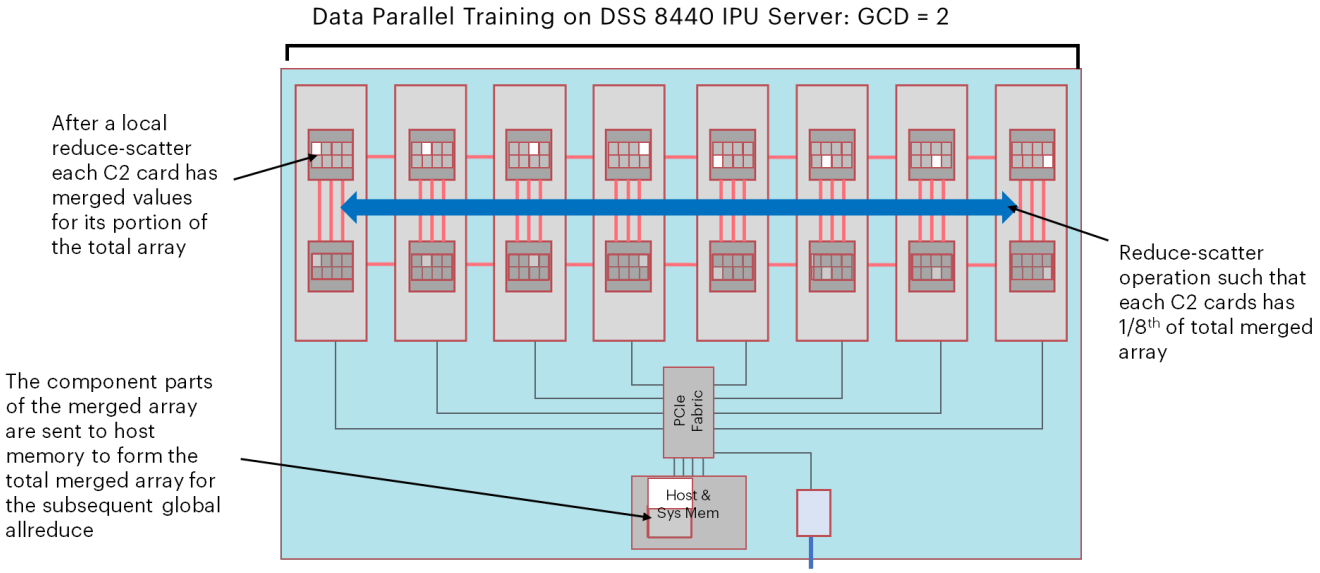
데이터 병렬 학습(Data Parallel Training) 또는 다중 복제 학습(Multiple Replica Training)

DNNs 학습의 첫 단계는 입력 데이터의 정방향 추론 계산(forward inference pass)입니다. 모든 모델 레플리카 전반의 입력 데이터 양은 미니배치(minibatch) 사이즈에 의해 결정됩니다; 정방향 계산 후 오류(loss)는 올바른 정답(correct answer)과 결과를 비교하여 계산됩니다. 이 loss는 역전파 단계(back propagation)에서 사용되어 모델 매개변수에 대한 가중치 그래디언트(weight gradient) 및 업데이트를 계산합니다. DSS 8440 IPU-서버 내 모델의 여러 레플리카를 이용한 학습(및 서버에서 지원되는

100Gbps 포트를 사용해 여러 DSS 8440 IPU-서버 전반에 확장하는 방식을 통한 학습)은 데이터 병렬(data parallel) 학습이며 각 모델 레플리카가 2대 이상의 IPU (GCD >1) 전반에 분산되어 있는 모델 병렬(model parallel)과 결합될 수 있습니다. 아래 그림은 GCD = 2 (C2 카드 내 2 IPU 전반의 각 모델 레플리카)인 데이터 병렬 학습의 예시입니다. 각 모델 레플리카가 병렬로 여러 다른 입력 데이터를 처리했기 때문에 역전파(back propagation)에서 계산된 그래디언트 배열(gradient array)은 각 케이스로 다를 것입니다(위 그림에서 각 "상단" IPU에 서로 다른 색의 박스로 표시됨).



목표는 '올리듀스(allreduce)' 프로세스를 이용해 모든 결과를 모든 모델 레플리카 전반의 단일 매개변수 업데이트로 결합하는 것입니다(아래 그림의 흰색 박스로 표시. 포플러 SDK는 DSS 8440 IPU-서버 내 여러 다른 올리듀스 구현을 지원합니다.



로컬 올리듀스 함수가 완료되면, 전역(global) 올리듀스 함수가 인피니밴드(InfiniBand) 지원을 통해 실행됩니다.

다수의 DSS 8440 IPU-서버 확장을 위한 인피니밴드(InfiniBand)

DSS 8440 IPU-서버는 고대역 및 짧은 지연시간의 서버 간 통신을 위해 IPU-Link를 사용하는 동시에 단일 100Gbps 인피니밴드(InfiniBand) 어댑터를 사용해 다수의 서버로 유연하게 스케일 아웃(scale-out)을 실행함으로써 총 200Gbps의 양방향 대역폭을 지원합니다.

요약

그래프코어는 지난 2년 간 업계 유수 기업들은 물론, 다수의 글로벌 혁신 선도 업체들과 협력해 학습 및 추론을 지원하는 IPU 프로세서와 포플러(Poplar) 소프트웨어 스택을 개발했습니다. 포플러는 혁신 선도 업체들이 텐서플로우, ONNX, 파이토치 등 일반적인 하이레벨 머신 러닝 프레임워크를 사용할 수 있도록 지원해 IPU에서 기존 모델을 손쉽게 활용할 수 있도록 돕고 있습니다.

즉시 사용 가능한 IPU는 딥러닝 모델에서 뛰어난 성능을 제공합니다. 그래프코어의 공개 벤치마크를 https://github.com/graphcore 에서 확인하시고 차세대 머신 인텔리전스를 구현하기 위한 그래프코어 IPU 도입을 검토하십시오.

그래프코어는 본 자료에서 제공된 정보가 발행일 현재 기준으로 정확하다고 판단합니다. 시간 경과 및 기술 발전에 따라 본 문서 내 정보는 시대에 맞지 않거나 불충분 또는 부정확해질 수 있습니다. 그래프코어 홈페이지를 통해 가장 최신 정보를 확인하십시오.

문의 사항

홈페이지: www.graphcore.ai

이메일: info@graphcore.ai