

기술 백서

REAL-WORLD AI

플래시블레이드(FLASHBLADE)를 활용한
딥러닝 가속화

목차

소개	3
데이터의 수명주기	3
데이터 과학자의 워크플로우.....	5
확장 가능한 데이터세트	6
왜 플래시블레이드(FlashBlade)인가.....	6
시스템 아키텍처	8
인프라 구성.....	8
소프트웨어 파이프라인	8
일반적인 훈련 파이프라인	9
네트워크 구성	10
벤치마크 결과.....	10
테스트 환경 구성	10
결과.....	11
성능 규모 산정 시 고려사항	13
대안: 로컬 스토리지(SSD)로의 데이터 스테이징	14
부록: 실 운영 시스템 아키텍처.....	15
물리적 인프라	16
플래시블레이드(FlashBlade) 구성환경.....	17
DGX-1 구성환경.....	17

소개

심층 신경망(Deep Neural Network, DNN)의 발전으로 데이터 과학자들이 인공지능 (AI)기반의 데이터를 활용할 수 있는 새로운 알고리즘 및 다양한 도구가 등장했습니다. 향상된 알고리즘, 대용량 데이터 세트 및 TensorFlow와 같은 프레임 워크를 사용하여 데이터 과학자는 자율 주행 자동차 및 자연 언어 처리와 같은 새로운 적용 사례를 만들고 있습니다.

심층 신경망(DNN)을 훈련(Training) 시키려면 고품질의 입력 데이터와 대규모의 계산 능력이 필요합니다. GPU는 방대한 병렬 프로세서 기반으로 대규모 데이터 연산을 동시에 수행할 수 있습니다. 여러 개의 GPU를 단일 클러스터로 결합하는 경우, 스토리지에서 컴퓨터 엔진으로 데이터를 공급하기 위해 고속의 데이터 파이프라인이 필요합니다. 딥 러닝은 모델을 구축하고 훈련시키는 것이 전부가 아닙니다. 데이터 과학자(팀)의 성공을 위해서는 확장성, 반복적인 실행, 그리고 실험에 필요한 데이터 파이프라인이 반드시 갖추어져야 합니다.

본 문서는 End-to-End 훈련 시스템이 가져오는 혜택과 그 기술적인 이유를 살펴보고, 왜 퓨어스토리지 플래시블레이드(FlashBlade™)가 이를 위한 핵심적인 플랫폼인지를 설명합니다. 또한 딥 러닝(Deep Learning) 애플리케이션을 위해 설계된 멀티 GPU 서버 NVIDIA® DGX-1™과 전체 AI 데이터 파이프라인을 위한 스케일 아웃기반의 고성능 다이내믹 데이터 허브인 플래시블레이드(FlashBlade)를 결합한 시스템의 성능 벤치 마크 결과를 보여줍니다.

데이터의 수명주기

데이터는 AI 및 딥러닝 알고리즘의 핵심입니다. 훈련 과정을 시작하기 전에, 먼저 정확한 AI 모델 훈련에 핵심이 되는 라벨링된 데이터를 수집해야 합니다. 대규모로 구현된 AI 시스템은 대용량의 데이터를 지속적으로 수집, 정제, 변환, 라벨링 및 저장해야 합니다. 더 많은 고품질 데이터를 통해 더 정확한 모델과 더 나은 통찰을 확보할 수 있습니다.

데이터 샘플들은 다음과 같은 일련의 처리 단계를 거칩니다.

- **INGEST:** 외부 소스로부터 훈련 시스템으로 데이터가 유입(Ingest)됩니다. 각 데이터는 보통 파일 또는 객체입니다. 이 데이터에 추론(Inference)을 실행할 수도 있습니다. 인제스트 단계 후, 데이터는 원시 형태로 저장되며 백업이 되기도 합니다. 모든 관련된 라벨(실측값)들이 데이터와 함께 또는 별도의 인제스트 스트림을 통해 유입됩니다.
- **CLEAN AND TRANSFORM:** 데이터를 정제(Clean) 및 변환(Transform)한 후, 데이터 샘플과 관련 라벨을 연동하고 훈련하기에 편리한 포맷으로 저장합니다. 이 단계에서 저장된 데이터는 필요한 경우 다시 계산하면 되기 때문에 백업되지 않습니다.
- **EXPLORE:** 파라미터와 모델을 찾고 이를 작은 규모의 데이터세트를 이용해 신속하게 테스트한 뒤, 가장 유망한 모델을 개발하여 운영 클러스터로 전달됩니다.
- **TRAINING:** 학습 단계에서는 입력 데이터의 새로운 샘플과 기존 샘플을 포함하여 무작위(Random)로 데이터가 선택되어 운영 GPU 서버로 보내지며 모델 파라미터 업데이트를 위한 학습이 수행됩니다.
- **EVALUATION:** 평가 단계에서는 학습에 사용되지 않은 데이터를 활용하여 모델의 정확성을 검증합니다.

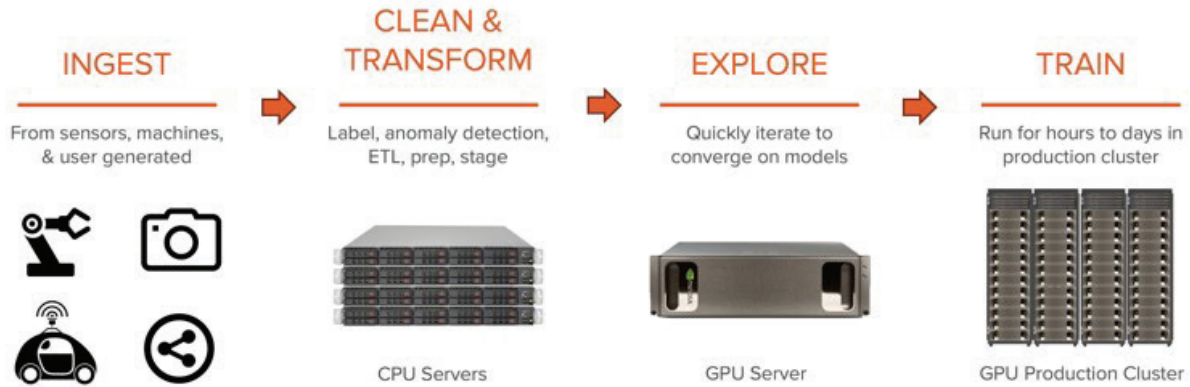


그림 1: AI 데이터 샘플 처리 단계

이러한 데이터 활용프로세스(Data Lifecycle)는 심층 신경망(DNN)이나 딥 러닝(Deep Learning)뿐만 아니라 모든 병렬처리 기반의 머신 러닝에 적용됩니다. 예를 들어, [Spark MLlib](#) 같은 표준 머신 러닝 프레임워크는 GPU가 아니라 CPU에 의존하지만, 데이터 학습 과정은 동일합니다.

공유 스토리지는 데이터 허브로 데이터 활용프로세스(Data Lifecycle) 전체에서 단일 저장소로서, 데이터 수집(유입, Ingest), 처리 및 훈련 단계들에서의 불필요한 데이터 이동을 방지합니다. 또한 인제스트된 데이터가 하나의 목적에만 사용되는 경우는 드물기 때문에 공유 스토리지는 데이터를 다양한 방법으로 해석하고, 다수의 모델을 훈련시키며, 데이터에 기존 분석을 적용할 수 있는 유연성을 제공합니다.

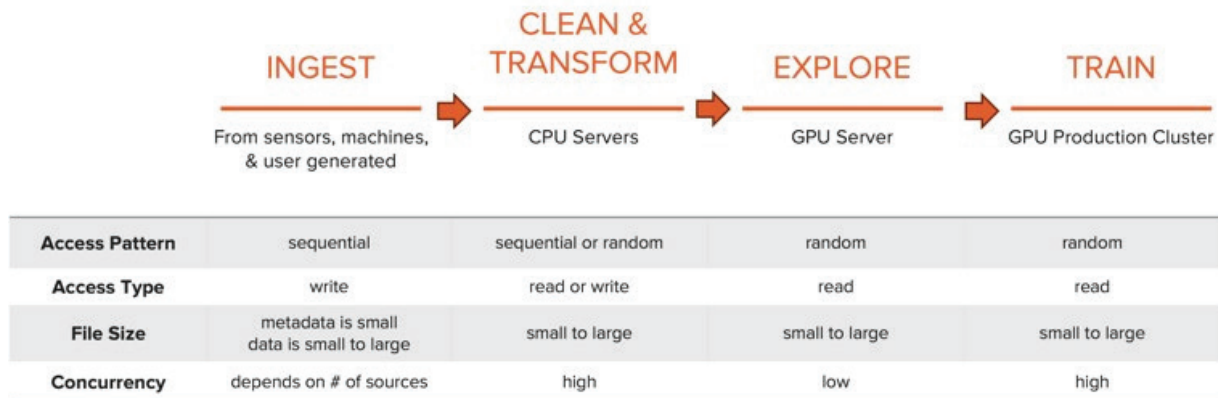


그림 2: AI 데이터 파이프라인에서의 스토리지 요구사항

위에서 보듯이, AI 데이터 파이프라인에서의 각 단계는 데이터허브에 다양한 요구사항을 필요로 합니다. 데이터 허브를 위한 수평확장(Scale-out) 스토리지 시스템은 작은 크기의 수많은 메타데이터, Small 데이터 및 Large 데이터 파일 활용은 물론, 랜덤 액세스 패턴과 순차적 액세스 패턴, 낮은 동시성과 높은 동시성 등, 모든 액세스 유형과 패턴에서 최고의 성능을 제공해야 합니다. 기존 레거시 스토리지 시스템은 이러한 요구조건을 위한 설계가 반영되어 있지 않습니다. 반면, 플래시블레이드(FlashBlade)는 AI를 위해 비정형데이터 환경에서의 최적화 설계를 통해 가장 이상적인 AI 데이터 허브를 구현할 수 있도록 하여 줍니다.

첫 번째 단계에서, 데이터는 적절하게 인제스트되어 동일한 데이터 허브에 저장되기 때문에 다음 단계들에서 과도한 데이터 복사가 필요하지 않습니다. 그 다음 두 단계들은 GPU를 옵션으로 포함하는 표준 컴퓨트 서버에서 수행될 수 있습니다. 그리고 네 번째 마지막 단계의 완전한 훈련 작업은 DGX-1과 같은 강력한 GPU 가속 서버에서 실행됩니다. 동일한 데이터세트를 활용한 운영 (Production) 파이프라인과 실험용(Experimental) 파이프라인이 공존하는 경우도 종종 있습니다. 이외에도, DGX-1 GPU들은 여러 다른 모델에서 독립적으로 사용되거나 함께 결합되어 더 큰 하나의 모델을 훈련시키기 위해 DGX-1 시스템들을 결합하여 분산 훈련(Distributed Training)을 하는 경우도 있습니다.

공유 스토리지가 느리면, 데이터를 각 단계에서 로컬 스토리지로 복사해야 합니다. 결과적으로 데이터를 다른 서버들로 스테이징 (Staging)하는데 시간이 허비됩니다. AI 데이터 파이프라인을 위한 이상적인 데이터 허브는 데이터가 시스템 메모리 (RAM) 에 저장된 경우와 유사한 성능을 제공하는 동시에 모든 파이프라인 단계들이 동시적으로 구동될 수 있도록 간단함과 성능을 제공해야 합니다.

데이터 과학자의 워크플로우

데이터 과학자는 더 많은 데이터, 더 나은 데이터, 더 스마트한 학습, 더 심도 있는 모델 등 광범위한 접근방식을 통해 학습된 모델의 정확성을 향상시키려고 합니다. 많은 경우, 데이터 과학자 팀들은 새롭고 향상된 훈련 모델을 생성하기 위해 동일한 데이터세트를 공유하며 병렬적으로 작업합니다.

데이터 과학자들의 일상적인 워크플로우는 다음과 같습니다.

- 데이터를 정렬, 정제, 필터링 및 처리하여 모델 훈련에 사용될 수 있는 형태로 변환
- 소규모 데이터 서브셋을 활용하여 모델을 실험, 테스트 및 디버깅
- 전체 훈련용 데이터세트로 모델 훈련

이러한 워크플로우는 세 단계 즉, 개발, 실험, 디버깅 단계를 반복합니다. 핵심적인 개발 툴은 [TensorFlow](#), [Caffe2](#), CNTK 같은 딥 러닝 프레임워크입니다. 이 프레임워크들은 데이터 처리 및 모델 구축을 위한 유틸리티를 제공합니다. 유틸리티들은 분산된 GPU 하드웨어에서의 실행에 최적화되어 있습니다.

대부분 데이터 과학자들은 팀을 이뤄, 동일한 데이터세트를 공유하며 이러한 단계들을 동시적으로 수행합니다. 데이터 처리, 실험 및 대규모 훈련을 다수의 사용자가 동시에 실행할 때 발생하는 워크로드는 스토리지에 다양한 액세스 패턴과 기능을 요구합니다. 다시 말해, 스토리지는 대규모의 파일 읽기만 충족시켜서 되는 것이 아니라 대소규모의 파일 읽기와 쓰기의 혼합을 처리할 수 있어야 합니다.

마지막으로, 다수의 데이터 과학자들이 데이터세트와 모델을 활용(Explorer)하는 경우 데이터를 원본형태로 저장하는 것이 중요합니다. 이는 각 사용자가 고유한 방식으로 데이터를 변환, 정제 및 사용할 수 있는 유연성을 제공합니다. 궁극적으로 이러한 실험들을 통해 보다 강력한 모델이 생성됩니다.

플래시블레이드(FlashBlade)는 데이터세트가 자연스럽게 공유되는 스토리지 허브를 제공합니다. 또한 다수의 개발자와 다수의 실험을 위한 공통 액세스 지점으로서의 역할을 수행하는데 필요한 성능과 데이터 보호(RAID6 활용)를 제공합니다.

플래시블레이드(FlashBlade)를 사용하면 로컬에서의 작업을 위해 데이터 일부(작업에 필요한 데이터)를 복사할 필요가 없기 때문에 DGX-1 시스템 낭비를 줄이고 엔지니어링 작업 시간이 감소됩니다. 불필요한 데이터 복사는 원시 데이터셋으로부터 변환이 지속적으로 업데이트되고 변경되기 때문에 지속적으로 증가하는 부담이 됩니다.

확장 가능한 데이터세트

딥 러닝의 성공률이 급격히 높아진 근본적인 이유는 더 큰 규모의 데이터셋을 통해 모델이 지속적으로 개선되기 때문입니다. 반대로, 로지스틱 회귀(Logistic Regression) 같은 기존 머신 러닝 알고리즘은 더 작은 규모의 데이터셋에서는 정확도가 향상되지 않습니다.

선두적인 AI 연구원의 말은 그 필요성을 강조해줍니다.

“2016년을 기점으로, Supervised 딥 러닝 알고리즘은 카테고리 별로 약 5,000개의 라벨링된 예시를 갖추어 허용할 만한 수준의 성능을 달성할 것이며, 라벨링된 예시가 1천만 개에 달하는 데이터셋으로 훈련 되면 인간의 능력과 유사하거나 능가할 것이다.”

[이안 굿펠로우\(Ian Goodfellow\) 2016년](#)

[Google의 최근 연구에 따르면](#) 데이터셋 크기가 증가함에 따라, 그리고 데이터셋이 3억 개의 이미지로 늘어나면서 이미지 작업의 성능이 크게 향상되는 것으로 나타났습니다. 이뿐만 아니라, 용량이 더 큰 모델은 이와 비례하여 더 큰 데이터셋을 필요로 한다는 사실이 밝혀졌습니다.

또한 컴퓨트(DGX-1)와 스토리지(FlashBlade)를 분리하면, [각 계층을 독립적으로 확장할 수 있기 때문에](#) 두 영역을 함께 관리함으로써 발생하는 다양한 복잡성을 제거할 수 있습니다. 데이터셋의 크기가 커지거나 새로운 데이터셋을 고려하면, 스케일 아웃 스토리지 시스템은 쉽게 확장될 수 있어야 합니다. 마찬가지로, 더 많은 동시적 훈련이 요구되면, 내부적인 스토리지에 대한 걱정 없이 더 많은 GPU 또는 DGX-1 서버를 추가할 수 있습니다.

왜 플래시블레이드(FlashBlade)인가

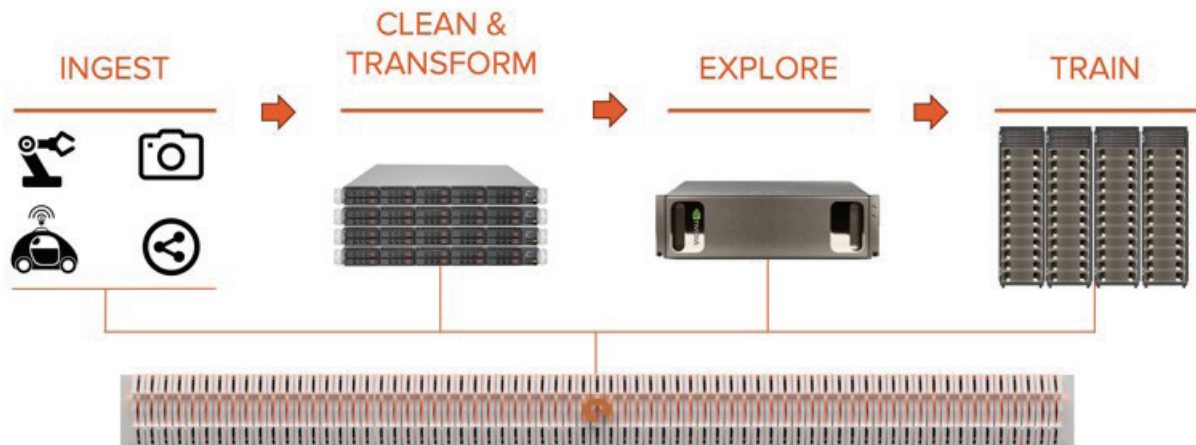


그림 3: 중앙화된 데이터 허브 FlashBlade

딥 러닝 아키텍처에서 중앙화된 데이터 허브는 데이터 과학자들의 생산성을 향상시켜주고, 데이터 아키텍처를 보다 단순하고 민첩하게 확장하고 운영할 수 있도록 해줍니다. 플래시블레이드(FlashBlade)는 AI 시스템을 보다 쉽게 구축, 운영 및 확장할 수 있도록 다음과 같은 혜택을 제공합니다.

- **Performance:** 읽기 대역폭이 새시당 15GB/초 이상이며 최대 75GB/초 (5개 FlashBlade 새시의 경우)인 플래시블레이드(FlashBlade)는 AI 워크플로우 전체에서 대용량의 동시 요구사항을 지원할 수 있습니다.
- **SMALL-FILE HANDLING:** 단일 FlashBlade 새시에서 무작위로 소형 파일(50KB)을 10GB/초의 속도로 읽을 수 있기 때문에(50GB/초, 블레이드 75개의 경우), 스토리지 성능에 맞춰 더 큰 크기의 파일을 만들기 위한 노력을 할 필요가 없습니다.
- **SCALABILITY:** 소규모 시스템으로 시작하고, 데이터세트 크기나 성능 요구사항이 증가하면 블레이드를 추가하여 용량과 성능을 증가시킬 수 있습니다.
- **NATIVE OBJECT SUPPORT(S3):** 파일이나 객체 기반의 데이터를 모두 저장/처리할 수 있습니다.
- **SIMPLE ADMINISTRATION:** 파일의 크기 및 개수 등에 따라 파일시스템 성능 튜닝을 고민하지 않아도 됩니다.
- **NON-DISRUPTIVE UPGRADE (NDU) EVERYTHING:** 모델을 훈련시키는 도중에도, 즉 언제나 소프트웨어와 하드웨어 업그레이드를 수행할 수 있습니다.
- **EASY OF MANAGEMENT:** 퓨어스토리지의 클라우드 관리 및 기술지원 플랫폼인 퓨어1(Pure1)은 사용자들의 모든 기기를 통해 (Phone, Tablet, Computer 등) 스토리지를 모니터링할 수 있도록 해주며, 예측적 지원(Predictive Support)을 통해 운영에 영향을 미치기 전 문제를 파악하고 해결합니다. 퓨어1(Pure1)은 사용자들이 스토리지 관리가 아니라 데이터를 이해하는데 주력할 수 있도록 해줍니다.
- **BUILT FOR THE FUTURE:** 플래시를 위해 특수 설계되어 새로운 세대의 NAND 기술(집적도, 비용 및 속도)을 손쉽게 활용할 수 있습니다.

스토리지는 텍스트, 오디오, 이미지뿐만 아니라 작은 파일로 저장되는 다양한 형태의 데이터에 대한 처리 성능이 매우 중요합니다. 스토리지에서 작은 파일을 효과적으로 처리(충분한 성능을 제공)하지 못하면, 사전 처리와 샘플들을 더 큰 파일로 그룹화해야 하는 등 추가적인 단계가 필요해집니다.

SSD를 캐시 영역으로 사용하는 기존 스토리지는 AI 환경에서 필요한 성능을 제공하지 못합니다. 무작위로 입력되는 다량의 데이터를 이용해 훈련시키게 되면 보다 더 정확한 모델을 생성할 수 있으므로, 전체 데이터세트는 빠른 성능 기반 하에 쉽게 액세스될 수 있어야 합니다. SSD 캐시는 소규모 데이터세트에서만 높은 성능을 발휘하며, 하드디스크 드라이브의 느린 응답속도를 개선하지 못합니다.

궁극적으로 플래시블레이드 (FlashBlade)의 성능과 동시처리 성능은 데이터 과학자로 하여금 불필요한 데이터 이동에 시간을 허비하지 않고 각 작업 단계 사이를 빠르게 이동할 수 있게 해줍니다. 또한 플래시블레이드(FlashBlade)는 동일한 데이터를 활용하여 여러 다른 작업을 동시에 수행할 수 있도록 해줍니다.

시스템 아키텍처

본 섹션에서는 딥 러닝 클러스터에 필요한 인프라와 소프트웨어 컴포넌트에 대해 알아보겠습니다.

인프라 구성

DGX-1과 플래시블레이드(FlashBlade)에 기반한 딥 러닝 클러스터는 다음의 하드웨어를 포함됩니다.

Primary Compute: 8 개의 V100 GPU가있는 DGX-1 서버는 심층신경망(DNN)을 학습하기 위해 상호 연결된 [NVLink](#)를 가지고 있습니다. 각 시스템에는 이더넷과 인피니밴드를 활용하여 외부 연결이 가능합니다. GPU는 단일 대규모 훈련(Training)을 위해 그룹화하거나 여러 모델을 훈련하기 위해 독립적으로 사용할 수 있습니다.

Storage: 플래시블레이드(FlashBlade)는 올플래시 기반으로 60TB에서 2.5PB까지 (비압축기준) 수평확장 가능하며 파일 및 객체 (Object)를 모두 저장할 수 있습니다. 고성능 NFS 또는 S3 프로토콜을 통해 데이터에 액세스할 수 있습니다.

Networking: 랙 상단의 이중화된 이더넷 스위치는 10Gbps 및 40Gbps MLAG 포트를 통해 스토리지와 컴퓨터에 연결되어 이중화를 제공합니다. DGX-1 시스템들은 보통 인피니밴드를 활용하여 서로 연결됩니다. 경우에 따라서 컴퓨터와 스토리지 트래픽 모두에 동일한 이더넷 스위치 패브릭을 사용하는 고객들도 있습니다.

Additional Compute: 데이터 수집, 사전 처리 및 모델 디버깅을 위해 선택적으로 GPU가있는 화이트 박스 서버가 있을 수 있습니다.

플래시블레이드(FlashBlade)와 DGX-1의 규모 산정은 “Sizing Guide” 섹션을 참조하십시오.

소프트웨어 파이프라인

훈련 소프트웨어는 스토리지 시스템에서 GPU로 데이터를 이동하여 GPU가 항상 데이터를 활용할 수 있도록 해주어야 합니다.

전체 모델 학습에서, 데이터 흐름은 다음과 같습니다.

- **DECODE AND ARGUMENT:** 스토리지에서 입력 파일을 로드하여 tensor 등 훈련에 적합한 형식으로 변환합니다. 디코딩 단계는 호스트 CPU에서 수행되거나 사전에 처리될 수 있습니다. 증강(Augmentation)은 호스트 CPU를 사용해서 해당 입력 데이터에 동적 변경을 적용하여 오버 피팅(과적합)을 회피합니다.
- **IO QUEUES:** 다수의 스레드는 스토리지로부터 랜덤 배치로 데이터를 읽은 후 내부 큐에 전달합니다. 이 스레드는 DGX-1 호스트 CPU에서 실행되고 사전 패칭 배치를 담당하며 DRAM에서 훈련 기록을 사용할 수 있도록 해줍니다.
- **TRAINING:** 두 번째 스레드 세트는 훈련과 모델 파라미터 업데이트에 필요한 계산을 할 수 있도록 내부 큐에서 데이터를 가져다가 GPU로 피드(즉, 전방향 연산 및 역전파)를 합니다.

각 훈련은 지속적으로 스토리지에서 데이터를 불러와 디코딩 및 증강해야 합니다. GPU가 유휴상태 즉, IO와 호스트 CPU가 입력을 가져와 디코딩해주길 기다리고 있는 상태인 경우에는 GPU가 효율적으로 활용되지 않습니다. 아래의 파이프라인에 있는 훈련 흐름은 훈련 장치가 하나의 훈련 배치가 완료되면 항상 다음 입력을 활용할 수 있도록 해줍니다. IO와 호스트 CPU 부분이 훈련 계산보다 빠른 경우, GPU는 항상 최대로 활용됩니다.

일반적인 훈련 파이프라인

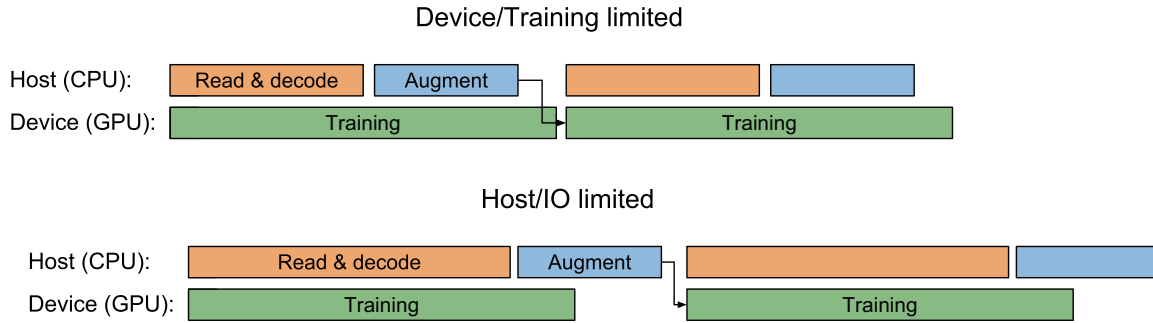


그림 4: 소프트웨어 아키텍처를 통한 훈련 데이터의 흐름

아래의 다이어그램은 앞서 설명된 파이프라인을 실현하기 위한 소프트웨어 아키텍처를 통한 훈련 데이터의 흐름을 보여줍니다. DGX-1의 호스트 CPU가 비동기화된 스레드 두 세트를 관리(운영)합니다. 페치(Fetch) 스레드는 랜덤하게 입력 파일을 선택하고, 가득 차지 않은 경우 큐로 데이터를 전달합니다. 훈련 스레드는 반복적으로 모델을 실행하고 각 반복에서 입력 데이터의 큐를 제거합니다. 이 두 프로세스 간의 조정 지점은 메모리 내 큐입니다. 페치 스레드는 입력 데이터를 미리 처리하여 예상된 형식으로 변환할 수도 있습니다. IO와 호스트 CPU의 처리속도가 큐가 비워지지 않을 만큼 되면, 훈련 스레드는 최적의 성능으로 실행될 수 있습니다.

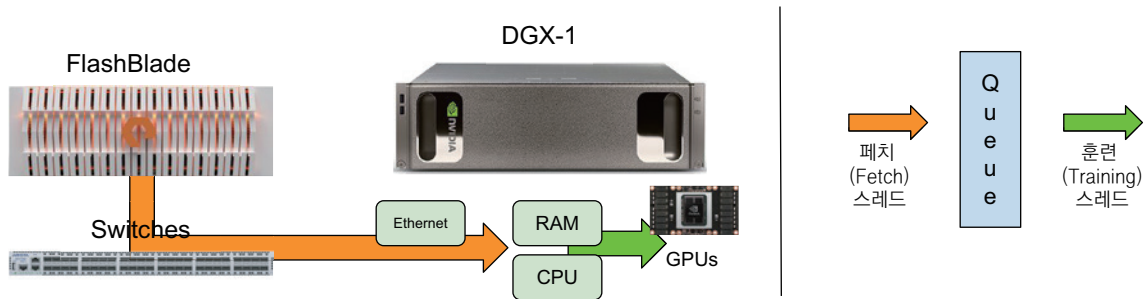


그림 5: FlashBlade의 속도는 최적의 훈련 성능을 지원합니다.

TensorFlow 문서에는 [read pipelining](#)에 대한 보다 세부적인 정보가 포함됩니다.

네트워크 구성

DGX-1은 인피니밴드(IB)와 이더넷 포트가 모두 있어 플래시블레이드(FlashBlade)와는 이더넷으로 연결이 됩니다. 플래시블레이드(FlashBlade)는 하나의 논리 링크를 제공하며 이는 단일 MLAG에서 최대 8x 40Gbps 포트를 통해 TOR 스위치에 연결이 됩니다. 모든 스토리지 액세스는 기본 마운트 옵션을 사용하여 NFS v3 프로토콜로 이루어집니다. rsize와 wsize는 512KB이고 파일 시스템 캐싱은 필요하지 않습니다.

DGX-1의 IB 포트는 대기 시간이 짧은 네트워크 기반의 여러 대 DGX-1 서버를 연결하므로 확장 가능한 훈련(Training) 능력을 제공하며, 또한 이더넷으로 연결하여 외부 스토리지와의 대역폭을 충분히 늘릴 수 있습니다. 내부 NVLink 연결은 여러 GPU를 상호 연결하며 각 훈련 반복 후에 모델 매개 변수에 대한 업데이트를 조정하는 데 사용됩니다. 여러 DGX-1으로 확장하기 위해 IB 포트를 사용하여 여러 대의 DGX-1 서버를 연결하여 활용할 수 있습니다.

가장 최신 모델의 GPU와 함께, 두 개의 10Gbps 이더넷 포트는 GPU가 데이터를 활용하는데 필요한 충분한 속도를 제공합니다. 단일한 10Gbps 링크의 대역폭 이상을 요구하는 훈련의 경우, DGX-1의 이더넷 인터페이스 설정 옵션은 두 가지가 존재합니다.

- 링크들을 결합(bonding)하여 스위치와 플래시블레이드(FlashBlade)로의 단일한 논리적 20Gbps 연결을 구성하고 제공하는 것입니다.
- 각 인터페이스를 서로 다른 서브넷에 할당하고 각각의 서브넷에서 두 개의 다른 플래시블레이드(FlashBlade) 데이터 VIP(가상 IP 주소)를 연결합니다.

두 경우 모두, 목표는 두 링크 전반에서 입력 데이터의 읽기를 다중화(Multiplexing)하여 이더넷 대역폭을 최대한 활용하는 것입니다.

인터페이스가 결합 또는 분리된 경우에도 각 훈련 세션은 단일한 TCP 연결을 생성하고 단일 파일시스템(마운트 포인트)으로 연결됩니다. 플래시블레이드(FlashBlade)에 있는 다수의 데이터 VIP를 활용하여 동일한 파일 시스템을 다수의 마운트 포인트에서 마운트하여 인터페이스 활용을 최대화 합니다. 이는 서로 다른 IP 엔드포인트로 연결이 되어, 서로 다른 TCP 연결을 생성하기 때문입니다.

벤치마크 결과

다음 섹션은 플래시블레이드(FlashBlade)와 DGX-1 시스템에서 얻어진 벤치마크 결과를 보여줍니다. 플래시블레이드(FlashBlade)의 빠른 데이터 전송(Direct Read)이 GPU를 포화상태로 만든다는 사실을 입증하기 위해 보편적인 ImageNet 모델들의 성능을 측정했습니다.

테스트 환경 구성

시스템 아키텍처에 설명된 것처럼, 테스트 환경은 DGX-1과 플래시블레이드(FlashBlade) 하드웨어로 구성되었습니다. DGX-1은 8대의 V100 GPU, 2대의 호스트 CPU(총 80 코어), 시스템 메모리 512GB로 구성되며, 훈련 데이터는 15개 블레이드로 이루어진 플래시블레이드(FlashBlade)에 저장되었습니다. 모든 테스트는 NVIDIA가 공급한 컨테이너(nvcr.io/nvidia/tensorflow v17.07)에서 수행되었습니다.

테스트들에서는 훈련을 위한 입력데이터로 딥 러닝 연구에서 가장 영향력이 있는 것 중의 하나인 Imagenet 2012를 사용했습니다. 이 데이터세트는 128만 개의 이미지로 구성되며 총 크기가 143GB에 달합니다. 라벨링된 jpeg 이미지들을 보다 큰 파일 크기(각 최대 135MB)로 합쳐 입력 데이터로 사용했습니다. NFS 마운트의 파일시스템 캐싱(fsc 옵션)은 꺼두었습니다. TensorFlow의 성능 최적화를 위한 모범 사례에 따라 초당 처리되는 이미지 수가 안정적인 상태에 도달할 때까지 각 모델을 훈련시켰습니다.

딥 러닝 커뮤니티에서 ImageNet 이 중요한 역할을 하기는 하지만, ImageNet에 있는 파일의 사이즈와 포맷이 실제 환경에서의 데이터 요구사항을 반영하지는 않습니다. ImageNet 의 테스트 데이터는 사전 처리가 되어 크기가 좀 더 큰 파일로 만들어져 있습니다. 이는 기본 포맷(> 1 MB)을 유지해 데이터를 완전하게 활용하여 빠르게 새로운 Prototype을 만들고자 하는 요구에 반하는 일입니다.

벤치마크를 위해, 훈련 데이터의 세 가지 다른 입력 소스를 비교했습니다.

- 합성(Synthetic) 데이터는 시스템 메모리에 인공적으로 생성된 데이터입니다. 관련된 스토리지 IO나 jpeg 포맷으로부터의 이미지 디코딩이 존재하지 않음을 의미합니다. 이 구성은 GPU 처리의 한계를 파악하여 GPU가 병목인지 여부를 테스트했습니다.
- 4개의 로컬 SSD가 DGX-1 박스 내부에 훈련 데이터를 저장합니다. 이 시나리오에서는 로컬 SSD들이 데이터세트를 위한 캐시로 사용됩니다. 데이터세트는 다른 외부 스토리지 보관소로부터 DGX-1으로 스테이징 될 필요가 있습니다. 이러한 접근 방식은 로컬 용량(현재 8TB) 이상으로는 확장되지 않습니다.
- 15개 블레이드로 구성된 플래시블레이드(FlashBlade)가 훈련 데이터의 소스입니다. 데이터는 다수의 TCP 연결을 이용해 플래시블레이드(FlashBlade)에서 읽기가 수행 됩니다. 플래시블레이드(FlashBlade)는 최대 2.5PB로 확장이 가능하며, 전체 데이터세트에 일관성 있는 성능을 제공합니다.

| 벤치마크 테스트의 목표는 외부 스토리지의 레이턴시가 병목을 야기하지 않는다는 사실을 입증하는 것입니다.

플래시블레이드(FlashBlade)에 연결하는데 DGX-1에 있는 두 개의 10Gbps 링크를 모두 활용하기 위해, 다수의 TCP를 연결해 입력 데이터에 액세스할 필요가 있었습니다. 플래시블레이드(FlashBlade) 파일 시스템을 두 개의 다른 데이터 VIP에 마운트하여 훈련 단계에서 각 마운트지점에 동시의 복합적인 파일 읽기가 진행되도록 하였습니다.

결과

두 개의 스토리지 옵션 즉, 내부 SSD와 플래시블레이드(FlashBlade)의 읽기 처리 능력을 비교하여 기준을 생성했습니다.

DGX-1에서 측정된 랜덤 데이터에 대한 읽기 처리 속도

- Local 4x RAID-0 SSD: 2.09 GB/초
- 플래시블레이드(FlashBlade): 2x 10Gbp에서 2.14 GB/초

두 개의 10Gbps 인터페이스를 사용한 경우, 로컬 스토리지와 플래시블레이드(FlashBlade)의 읽기 대역폭은 유사했습니다. 이러한 2GB/초의 읽기 성능은 두 구성에서 최대 IO 성능이며, 컴퓨터는 유휴 성능을 통해서 데이터 인제스트, 실험 및 모델 디버깅 등에 활용합니다.

TensorFlow 벤치마크 결과

훈련에서 초당 처리되는 이미지 수를 안정적으로 측정하기 위해 TensorFlow 벤치마크를 실행했습니다. SGD 최적화된 데이터 병렬 모드에서 다양한 수의 GPU로 각 벤치마크를 실행했습니다.

아래의 차트에서 이미지/초 속도가 높은 부분은 수직 축으로 표시되며, 프로세싱 속도와 처리요구량이 보다 높음을 나타냅니다. 속도는 처리 속도가 아니라 이미지/초로 측정이 되었습니다. 각 모델에는 기본 입력 이미지 사이즈에 상관없이 고정된 사이즈 입력이 있어야 하기 때문입니다. 각 파이프라인에는 데이터가 GPU로 전송되기 전 리사이징되는 단계가 포함됩니다. 크기가 큰 입력 파일은 스토리지의 부하를 증가시키지만 GPU에서는 그렇지 않습니다.

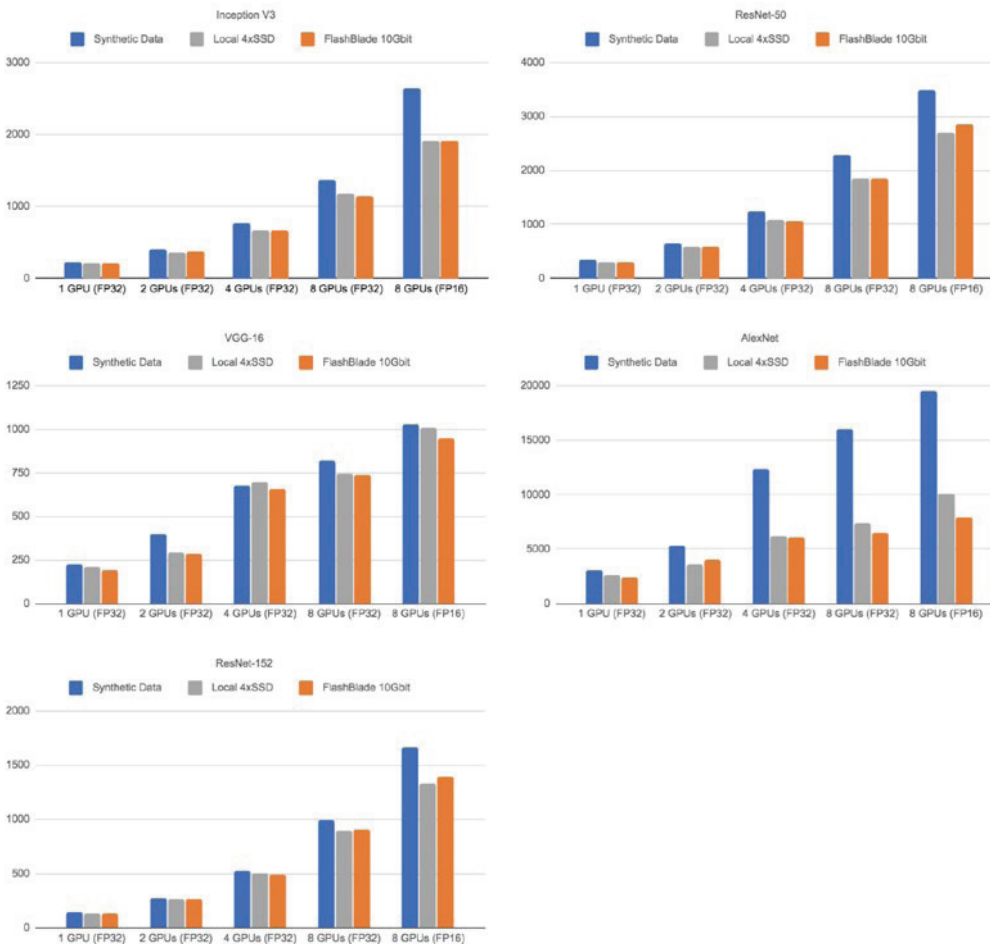


그림 6: 모델별 TensorFlow 벤치마크 결과 AlexNet는 이미지 변환 때문에 CPU 바운드입니다. 모든 다른 모델들은 GPU 바운드입니다.

보다 복잡한 모델인 Inception-v3, ResNet-50, Resnet-152 및 VGG-16은 모두 GPU 병목을 야기했습니다. 이는 합성 데이터와 플래시블레이드(FlashBlade) 훈련 성능이 거의 동일하다는 사실로 입증됩니다. 인메모리 큐가 GPU의 처리 성능 수요를 충족시킬 만큼 충분히 가득 찬 상태로 유지되었음을 의미합니다. 합성 워크로드는 로컬 SSD와 플래시블레이드(FlashBlade) 보다 약간 빨랐습니다. 핵심적인 경로의 작은 부분을 차지하는 CPU에서의 이미지 변환 탓입니다.

반대로, DGX-1의 훈련 전 수행된 이미지 변환 때문에 AlexNet은 CPU 중심 연산으로 나타났습니다. 합성 데이터를 사용하여, 훈련에서는 초당 20k 이미지를 처리했으며, IO 는 최대 약 9,200 이미지/초, 또는 약 1.2GB/초를 보였습니다. 이 속도는 스토리지의 측정된 읽기 성능 보다 낮아 외부 스토리지 IO가 아니라 이미지 파일 처리가 병목임을 시사해줍니다.

성능 규모 산정 시 고려사항

적절한 플래시블레이드(FlashBlade) 크기는 1) 훈련되는 모델의 복잡성과 2) 입력 데이터의 기본 크기 그리고 동시에 운영되는 GPU의 수에 따라 달라집니다. 세 번째 요소는 팀에 클러스터 리소스를 사용하는 데이터 과학자가 몇 명이 있는가와 직결됩니다. 각 요소들은 다른 파이프라인 단계 이외에도 필요한 성능과 딥 러닝을 지원하는 데 필요한 블레이드 수에 영향을 줍니다.

- 보다 복잡한 모델들은 각 훈련 예시에서 더 많은 계산을 필요로 하고, 결과적으로 보다 간단한 모델들에 비해 읽기 처리 역량이 덜 필요합니다.
- 크기가 큰 파일은 입력 단계에서 스토리지 계층으로부터 더 많은 읽기 처리량이 요구됩니다. 모델의 복잡성과는 상관이 없다는 사실에 주목하십시오. 이미지들은 훈련 이전에 리사이징됩니다.
- 추가적인 GPU 서버들은 다수의 독립 모델들이 동시에 훈련되도록 하거나 단일한 모델 훈련이 수평 확장 되도록 하여 로드를 증가시킵니다. 향후 더 빠른 GPU가 나오면 데이터에 대한 수요 역시 증가할 것입니다.

기본 입력 사이즈(Native Input Size)를 사용하는 것은 데이터 과학자들이 데이터를 가장 유용한 형태로 변환 및 처리할 수 있는 가장 큰 유연성을 제공합니다.

중요한 점은 입력 사이즈와 모델의 복잡성 간의 관계입니다. 예를 들어, ResNet-50과 같은 ImageNet 모델은 입력을 225x225 이미지로 조정합니다. 훈련 전에 모든 이미지를 이 크기로 사전 처리하면 처리량이 줄어들지만 이는 어떻게 데이터가 사용될지 고정된 추정을 강요하는 것입니다. 더 큰 입력 사이즈를 활용하기 위해 더 큰 모델들을 생성하는 것은 입력 수가 더불어 확장되지 않는다면 효과가 없습니다. 입력점 보다 더 많은 모델 파라미터가 존재한다면, 모델은 학습하는 것이 아니라 훈련 데이터를 단순히 암기해 버릴 것이기 때문입니다.

크기 산정에 대한 지침을 제공하기 위해, 아래 표는 복잡성과 사이즈 면에서 큰 편차가 존재하는 가장 잘 알려진 ImageNet 모델들을 보여줍니다. 8-GPU 훈련의 측정된 처리량은 네이티브 입력 사이즈 150KB를 사용하고, 외삽된 처리 역량 요구는 또한 1MB 크기 이미지들에 포함이 됩니다. 플래시블레이드(FlashBlade)가 데이터를 제공할 수 있는 DGX-1 시스템의 수를 산정하기 위해, 가용 처리 역량의 50%는 훈련에 사용되고, 다른 50%는 인제스트, 변환 및 실험에 사용된다고 가정을 합니다.

IMAGENET Model	DGX-1 Read Throughput		# of DGX-1 SYSTEMS PER FLASHBLADE	
	150KB Image size From IMAGENET	1MB Image Size , Extrapolate	7-BLADE SYSTEM, USING 1MB IMAGES	15-BLADE SYSTEM, USING 1MB IMAGES
INCEPTION V3	290 MB/초	1.90 GB/초	2	4
RESNET-50	430 MB/초	2.90 GB/초	1	2
RESNET-152	210 MB/초	1.40 GB/초	2	5
ALEXNET	1200 MB/초	8.00 GB/초	–	1
VGG16	140 MB/초	1.00 GB/초	3	7

플래시블레이드(FlashBlade) 성능은 입력 파일들의 크기에 좌우됩니다.

- 작은 사이즈 파일(50KB)의 경우, 7개 블레이드로 구성된 플래시블레이드(FlashBlade) 시스템은 약 5GB/초의 읽기 성능을 낼 수 있으며 15개 블레이드 시스템은 10GB/초가 넘습니다.
- 큰 사이즈 파일(1MB 이상)의 경우, 읽기 성능은 블레이드 당 1GB/초를 약간 상회하여, 15개 블레이드 시스템의 경우 15GB/초의 성능을 발휘합니다.

이미지들이 파일당 1MB의 기본 포맷으로 유지된 ResNet-50 모델의 경우, 요구되는 성능은 2.9 GB/초입니다. 두 개의 동시적 훈련에는 약 6GB/초의 읽기 처리 역량이 필요하여, 데이터 인제스트, 데이터 처리, 모델 디버깅 및 테스트를 위한 성능에 여유가 있습니다.

앞으로는 훈련 속도를 가속화할 수 있는 보다 강력한 GPU가 출시되어 스토리지 시스템 성능에 대한 수요가 증가될 것입니다. 더 강력한 GPU는 더 깊은 신경망을 가능하게 하여 입력 배치당 계산 양을 증가시켜 줍니다. 보다 심도 있는 모델들로 인해 필요한 성능 증가분은 일부 상쇄될 것입니다.

대안: 로컬 스토리지(SSD)로의 데이터 스테이징

소형 데이터세트의 경우 고성능 공유 스토리지에 대한 대안은 데이터를 DGX-1에 있는 로컬 SSD로 스테이징하는 것입니다. 훈련 세트가 변경될 필요가 있을 때마다, 사용자는 새로운 데이터를 스테이징해야 합니다. 로컬 스토리지보다 큰 데이터세트의 경우, 이러한 접근방식은 사용할 수 없습니다.

예를 들어, 200kB 이미지로 구성된 1TB의 데이터를 DGX-1 SSD로 스테이징하는 데는 1.5~2.5시간이 걸립니다. 이러한 스테이징은 DGX-1에서 GPU가 사용되지 않는 시간을 의미합니다. 해결에 걸리는 총 시간을 측정하는 경우, 공유 스토리지가 현저히 더 빠른 종단간 성능을 제공할 수 있습니다.

부록: 실 운영 시스템 아키텍처

실 운영 시스템은 딥 러닝에 필요한 풍부한 데이터셋을 저장하고 다양한 모델 훈련을 위해 GPU서버를 활용합니다. 아래는 자율주행 SW개발을 위해 구현된 딥 러닝 시스템의 실 환경 아키텍처입니다.

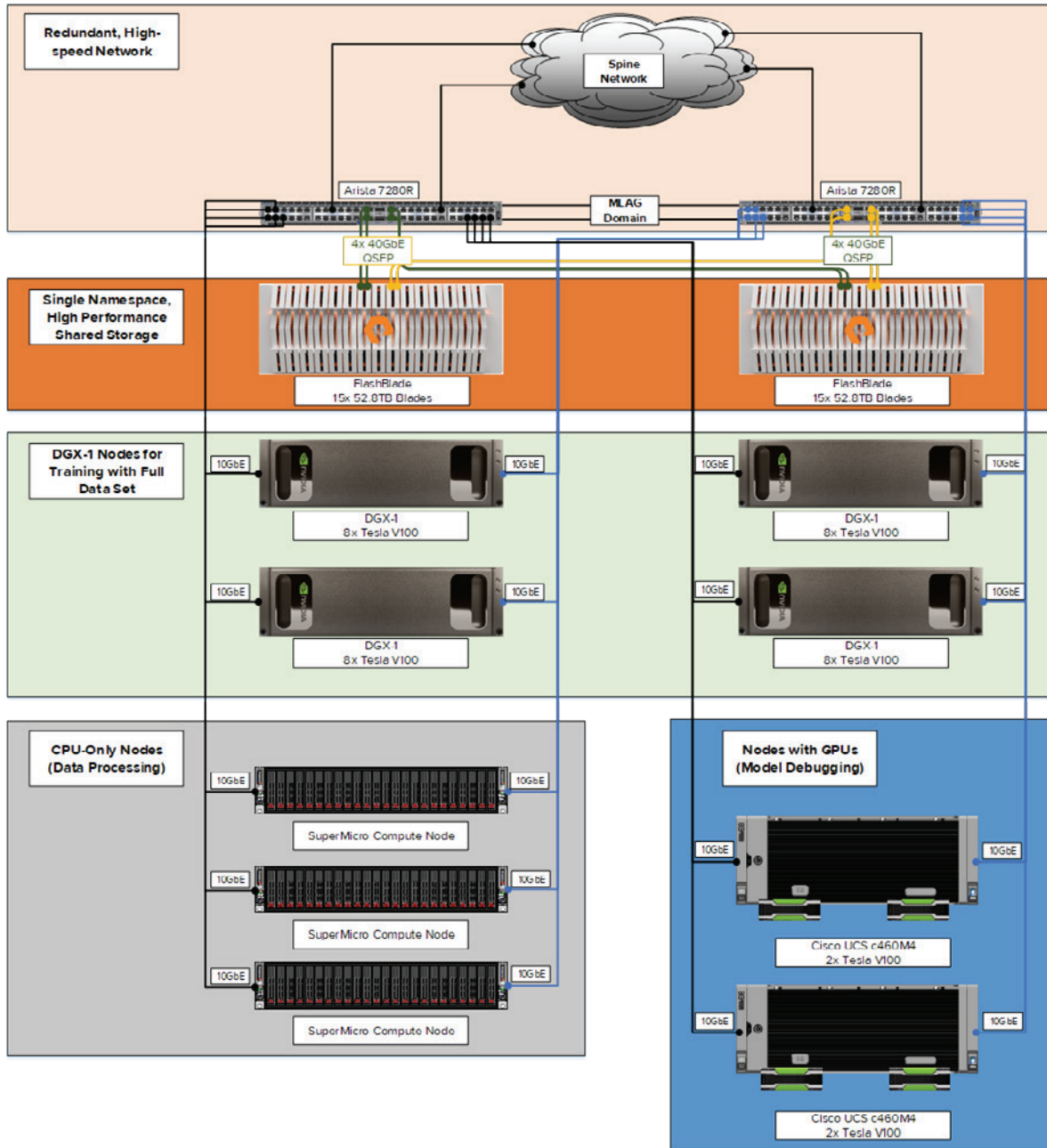


그림 7: 완전한 시스템 아키텍처의 예시

물리적 인프라

그림 7은 아래의 구성요소들로 이루어져 있습니다.

Primary Compute: DGX-1 (3U)

- Nvidia V100 GPU 8대
- L2기반 이더넷 스위치를 통해 플래시블레이드(FlashBlade)와 DGX-1 연결
- 플래시블레이드(FlashBlade) 데이터 VIP와 동일한 서브넷 설정

Storage: FlashBlade (4U)

- 성능 요구사항에 따라 블레이드 7~15대
- 데이터세트 크기에 따라 17TB 또는 52TB 블레이드
- 17TB 블레이드는 52TB 블레이드 보다 3배 높은 성능 제공(가용용량 TB기준, MB/초로 측정)
- 다수 데이터 VIP와 단일한 파일 시스템 구성

Additional Compute Server

- 데이터 전 처리, 정제 및 정규화등 수행
- 소규모 NN(Neural Network) 모델 검증
- 모델 훈련 및 디버깅을 해주는 추가적인 GPU 지원 서버

Networking: TOR Switch pair in MLAG

- DGx-1의 각 네트워크 포트는 안정성을 위해 서로 다른 TOR 스위치에 연결
- DGX-1 서버 네트워크 구성 예(DGX-1-1 및 DGX-1-2):
- DGX-1-1.port1 -> switch1
 - DGX-1-1.port2 -> switch2
 - DGX-1-2.port1 -> switch1
 - DGX-1-2.port2 -> switch2
- 플래시블레이드(FlashBlade)의 각 네트워크 포트는 안정성을 위해 서로 다른 TOR 스위치에 연결
 - 연결: **FM1** -> switch1, FM1 -> switch2, **FM2** -> switch1, **FM2** -> switch2

플래시블레이드(FlashBlade) 구성환경

훈련(Training) 데이터를 위한 단일 파일 시스템 구성

PUREUSER@SN1-FB-E02-33-1:~\$ PUREFS LIST							
NAME	SIZE	USED	% USED	CREATED	PROTOCOL	RULE	FAST REMOVE
NFS0	1T	138.40G	14%	2017-08-03 14:08:04 PDT	NFS	*(RW,NO_ ROOT_ SQUASH)	FALSE

플래시블레이드(FlashBlade) 다수의 데이터 VIP 활용

PUREUSER@SN1-FB-E02-33-1:~\$ PURENETWORK LIST							
NAME	ENABLED	SUBNET	ADDRESS	VLAN MASK	GATEWAY	MTU	SERVICE
NFS	TRUE	NET3	10.21.115.9	2115	10.21.115.1	1500	DATA
NFS0	TRUE	NET3	10.21.115.6	2115	10.21.115.1	1500	DATA
NFS1	TRUE	NET3	10.21.115.7	2115	10.21.115.1	1500	DATA
NFS2	TRUE	NET3	10.21.115.8	2115	10.21.115.1	1500	DATA
VIRO	TRUE	NET2	10.21.112.113	2112	10.21.112.1	1500	MANAGEMENT

DGX-1 구성환경

FLASHBLADE Mount설정

```
pureuser@sn1-dgx-1-e02-37:~/tensorow-benchmarks$ cat /etc/mtab
/dev/sda2 / ext4 rw,errors=remount-ro 0 0
proc /proc proc rw,noexec,nosuid,nodev 0 0
sysfs /sys sysfs rw,noexec,nosuid,nodev 0 0
.....
/dev/sda1 /boot/efi vfat rw 0 0
/dev/sdb1 /raid ext4 rw 0 0
rpc_pipefs /run/rpc_pipefs rpc_pipefs rw 0 0
systemd /sys/fs/cgroup/systemd cgroup rw,noexec,nosuid,nodev,none,name=systemd 0 0
:/nfs0 /mnt/nfs0 nfs rw,addr=10.21.115.6,_netdev 0 0
:/nfs0 /mnt/nfs1 nfs rw,addr=10.21.115.7,_netdev 0 0
:/nfs0 /mnt/nfs2 nfs rw,addr=10.21.115.8,_netdev 0 0
:/nfs0 /mnt/nfs3 nfs rw,addr=10.21.115.9,_netdev 0 0
```

Routing Table to Split the Mount Point Traffic Across Both Interfaces

```
pureuser@sn1-dgx-1-e02-37:~/tensorflow-benchmarks$ ip route
default via 10.21.115.1 dev em1
10.21.115.0/24 dev em1 proto kernel scope link src 10.21.115.120
10.21.115.6 dev em1 proto kernel scope link src 10.21.115.120
10.21.115.7 dev em2 proto kernel scope link src 10.21.115.113
10.21.115.8 dev em1 proto kernel scope link src 10.21.115.120
10.21.115.9 dev em2 proto kernel scope link src 10.21.115.113
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1
```

© 2018 Pure Storage, Inc. All rights reserved.

Pure Storage, FlashBlade, 및 Pure Storage 로고는 미국 또는 그 외 국가에 위치한 자회사의 상표 또는 등록상표입니다. Nvidia와 DGX-1은 Nvidia, Inc.의 등록상표입니다. 기타 기업, 제품 또는 서비스 이름은 해당 기업의 등록상표입니다.

이 문서에 설명된 퓨어스토리지 제품들은 제품의 사용, 복사, 배포 및 역컴파일/역엔지니어링을 제한하는 라이선스 계약 하에 배포됩니다. 이 문서에 설명된 퓨어스토리지 제품들은 라이선스 계약의 조건에 따라서만 사용될 수 있습니다. 이 문서의 어떠한 부분도 퓨어스토리지의 사전 서면 허가 없이 어떠한 형식이나 방법으로든 복제될 수 없습니다. 퓨어스토리지는 이 문서에 포함된 퓨어스토리지의 제품 및/또는 프로그램을 사전 통지 없이 언제든지 임의대로 개선 및/또는 변경할 수 있습니다.

이 문서는 "있는 그대로" 제공되며, 퓨어스토리지는 법적으로 허용된 범위 내에서 상품성, 특수 목적을 위한 적합성, 또는 비침해성에 대한 보증은 물론 그 어떠한 명시적, 묵시적, 서면, 구술 또는 법적 보증을 부인합니다. 퓨어스토리지는 이 문서의 이용, 공급 또는 성과와 관련하여 발생하는 모든 우발적 또는 결과적 손해에 대해 어떠한 경우에도 책임을 지지 않습니다. 이 문서의 정보는 예고 없이 변경될 수 있습니다.

ps_wp18p_AI-reference-architecture_ltr_02

korea@purestorage.com | 02-6001-3330 | purestorage.com/kr

