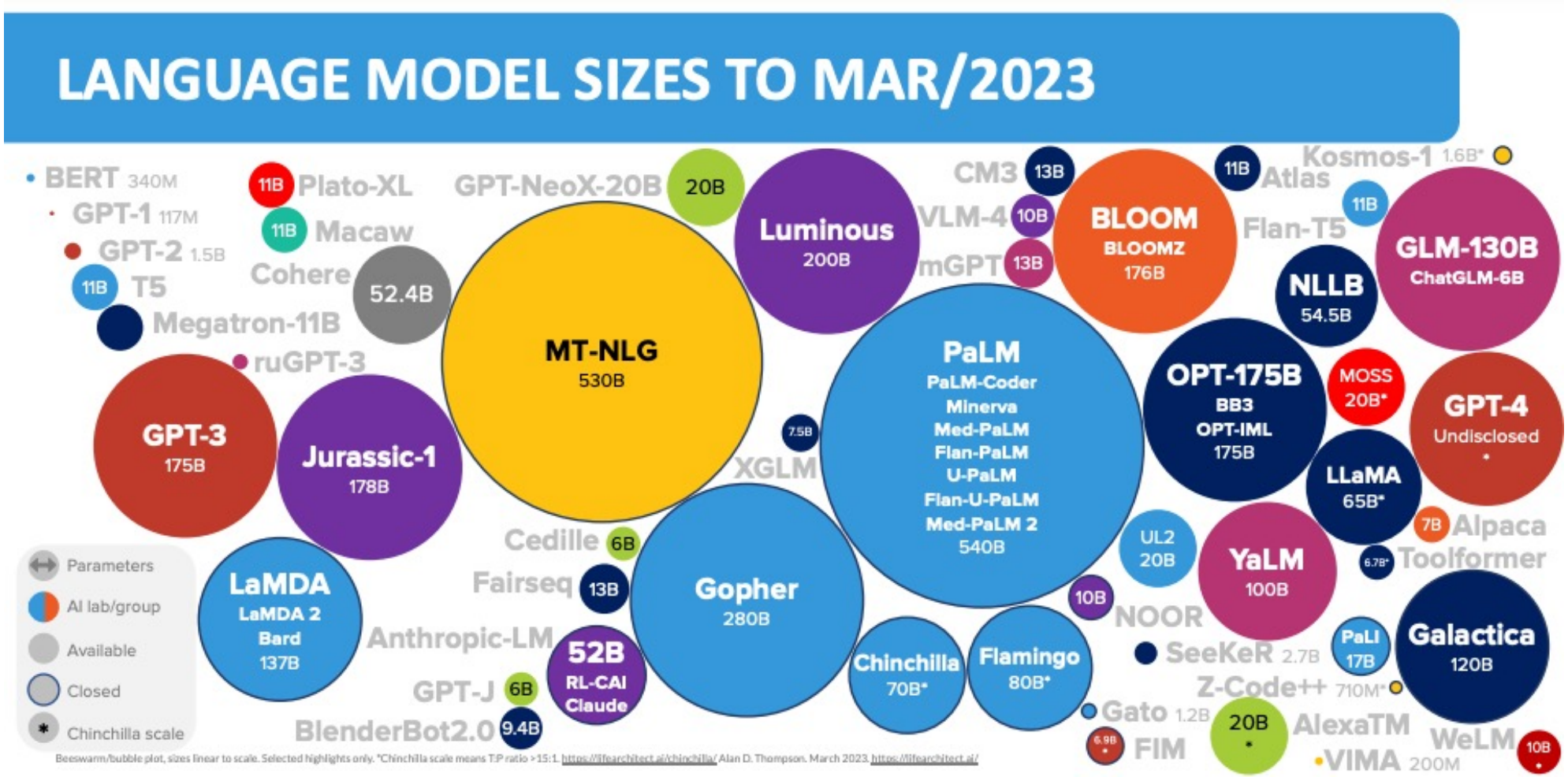
Optimizing Supercomputer for Large-Scale Training

김승식
SK Telecom

The golden age of large language models



https://lifearchitect.ai

모델의 정량적인 측정

- FLOPS, FLOPs 용어 정리
  - FLOPS (**FL**oating point **O**perations **P**er **S**econd)
    - 1초당 얼마나 많은 부동 소수점 연산을 처리할 수 있는가?
    - Hardware의 성능 지표
  - FLOPs (**FL**oating point **OP**eration**s**)
    - 부동 소수점 연산의 횟수
    - 모델의 복잡성과 연산량을 측정

Model	Top-1 Accuracy	GFLOPs	Year
AlexNet [Krizhevsky et al., 2012]	56.52	1.42	2012
ZFNet [Zeiler and Fergus, 2013]	60.21	2.34	2013
GoogleLeNet [Szegedy et al., 2014]	69.77	3.00	2014
MobileNet [Howard et al., 2017]	70.6	1.14	2017
MobileNetV2 1.4 [Sandler et al., 2019]	74.7	1.18	2018
EfficientNet-B1 [Tan and Le, 2020]	79.1	1.40	2019
NoisyStudent-B1 [Xie et al., 2020]	81.5	1.40	2019

	A100 40GB PCIe	A100 80GB PCIe	A100 40GB SXM	A100 80GB SXM
FP64	9.7 TFLOPS			
FP64 Tensor Core	19.5 TFLOPS			
FP32	19.5 TFLOPS			
Tensor Float 32 (TF32)	156 TFLOPS | 312 TFLOPS*			
BFLOAT16 Tensor Core	312 TFLOPS | 624 TFLOPS*			
FP16 Tensor Core	312 TFLOPS | 624 TFLOPS*			
INT8 Tensor Core	624 TOPS | 1248 TOPS*			
GPU Memory	40GB HBM2	80GB HBM2e	40GB HBM2	80GB HBM2e
GPU Memory Bandwidth	1,555GB/s	1,935GB/s	1,555GB/s	2,039GB/s

GPT-3 175B 모델은 얼마만큼의 연산량이 필요할까?

- Iteration 당 FLOPs

$$96Bslh^2\left(1+\frac{s}{6h}+\frac{V}{16lh}\right)$$ ①

B : Batch size
s : sequence length
l : transformer layer
h : hidden size
V : vocabulary size

$$96 * 1024 * 2048 * 96 * 12288^2\left(1+\frac{2048}{6*1288}+\frac{51200}{16*96*12288}\right) = 3.00E{+}18 = \boldsymbol{3\ ExaFLOPs}$$

- 10 만 Iteration을 학습 한다면

$$100000 * 3\ ExaFLOPS = 300\ ZettaFLOPs$$

- NVIDIA A100 GPU(312 TFLOPS, FP16) ② “1” 개로 학습을 한다면

$$300\ ZettaFLOPs\ /\ 312\ TeraFLOPS = 11{,}156\ days = \boldsymbol{30.5\ years}$$

1) Deepak Narayanan et al., Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM
2) https://www.nvidia.com/en-us/data-center/a100/

OPEN Compute Project®

GPT-3 175B 모델 학습 시간은 얼마나 걸릴까?

- End-to-end training time

$$\approx \frac{8TP}{nX}$$

T : Tokens
P : Model Parameters
n : GPUs
X : Achieved teraflop/s per GPU

①

- 175B 모델 학습 시간 (300B Tokens, A100 1040개)

$$8 * \frac{300 * 10^9 * 175 * 10^9}{1040 * 140 * 10^{12}} \approx \mathbf{34\ days}$$

- 7.5B 모델 학습 시간 (300B Tokens, A100 256개)

$$8 * \frac{300 * 10^9 * 7.5 * 10^9}{256 * 142 * 10^{12}} \approx \mathbf{6\ days}$$

Number of parameters (billion)	Attention heads	Hidden size	Number of layers	Tensor model-parallel size	Pipeline model-parallel size	Number of GPUs	Batch size	Achieved teraFLOP/s per GPU	Percentage of theoretical peak FLOP/s	Achieved aggregate petaFLOP/s
1.7	24	2304	24	1	1	32	512	137	44%	4.4
3.6	32	3072	30	2	1	64	512	138	44%	8.8
7.5	32	4096	36	4	1	128	512	142	46%	18.2
18.4	48	6144	40	8	1	256	1024	135	43%	34.6
39.1	64	8192	48	8	2	512	1536	138	44%	70.8
76.1	80	10240	60	8	4	1024	1792	140	45%	143.8
145.6	96	12288	80	8	8	1536	2304	148	47%	227.1
310.1	128	16384	96	8	16	1920	2160	155	50%	297.4
529.6	128	20480	105	8	35	2520	2520	163	52%	410.2
1008.0	160	25600	128	8	64	3072	3072	163	52%	502.0

1) Deepak Narayanan et al., Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM

GPT-3 175B 모델 학습 시간은 얼마나 걸릴까?

- End-to-end training time

$$\approx \frac{8TP}{nX}$$

T : Tokens
P : Model Parameters
n : GPUs
X : Achieved teraflop/s per GPU

①

- 175B 모델 학습 시간 (300B Tokens, A100 1040개)

$$8 * \frac{300 * 10^9 * 175 * 10^9}{1040 * 140 * 10^{12}} \approx \mathbf{34\ days}$$

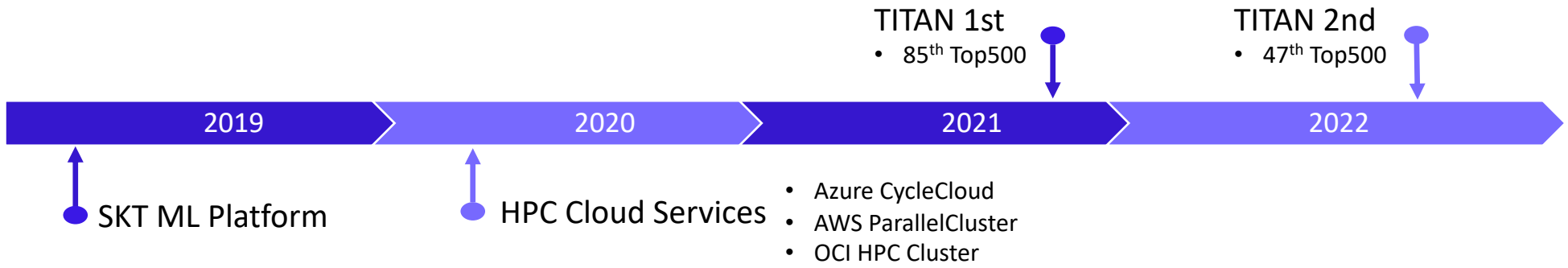
- 7.5B 모델 학습 시간 (300B Tokens, A100 256개)

$$8 * \frac{300 * 10^9 * 7.5 * 10^9}{256 * 142 * 10^{12}} \approx \mathbf{6\ days}$$

Number of parameters (billion)	Attention heads	Hidden size	Number of layers	Tensor model-parallel size	Pipeline model-parallel size	Number of GPUs	Batch size	Achieved teraFLOP/s per GPU	Percentage of theoretical peak FLOP/s	Achieved aggregate petaFLOP/s
1.7	24	2304	24	1	1	32	512	137	44%	4.4
3.6	32	3072	30	2	1	64	512	138	44%	8.8
7.5	32	4096	36	4	1	128	512	142	46%	18.2
18.4	48	6144	40	8	1	256	1024	135	43%	34.6
39.1	64	8192	48	8	2	512	1536	138	44%	70.8
76.1	80	10240	60	8	4	1024	1792	140	45%	143.8
145.6	96	12288	80	8	8	1536	2304	148	47%	227.1
310.1	128	16384	96	8	16	1920	2160	155	50%	297.4
529.6	128	20480	105	8	35	2520	2520	163	52%	410.2
1008.0	160	25600	128	8	64	3072	3072	163	52%	502.0

1) *Deepak Narayanan et al., Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM*

Supercomputer TITAN



- GPU Compute Nodes : 130
- NVIDIA A100 80G : 1,040
- InfiniBand HDR Switchs : 68
- InfiniBand Cables : 1,577

47위, TOP500 - JUNE 2023

47	**Titan** - Apollo 6500, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 80 GB, Mellanox HDR Infiniband, HPE SK Telecom South Korea	128,960	14.24	16.39

TOP 500 The List.

①

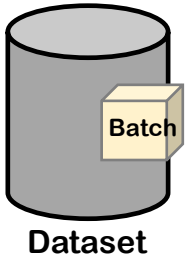
1) https://www.top500.org/lists/top500/list/2023/06/

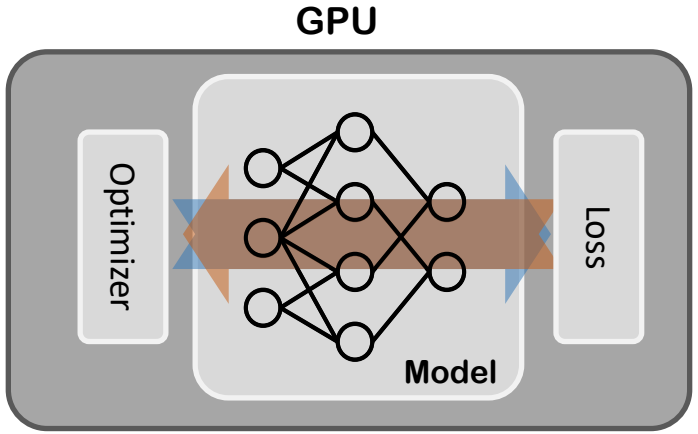
OPEN Compute Project®

Introduction to distributed training

Deep Neural Network 학습

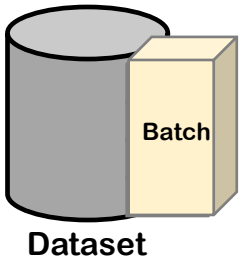
- DNN 학습 단계
  - Forward propagation
  - Loss
  - Backpropagation
  - Update weight

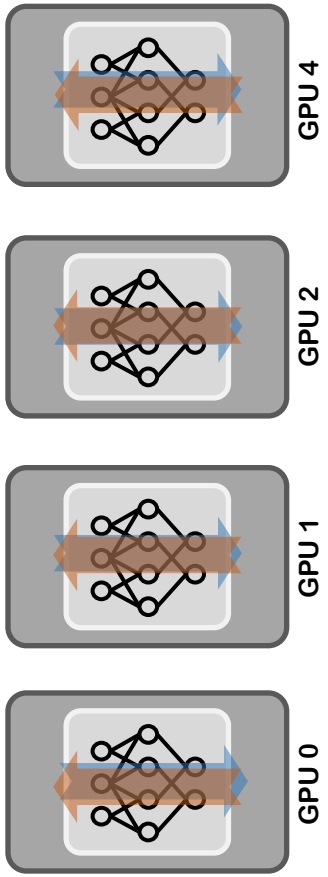




Data Parallelism

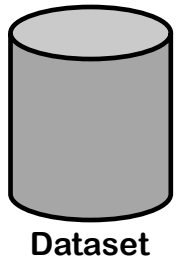
- **데이터 병렬화 기법의 학습 단계**
  - 전체 Dataset을 여러 개의 작은 Batch로 분할
  - 모델을 각각의 GPU에 복제
  - 각 GPU는 분할된 dataset을 통해 학습

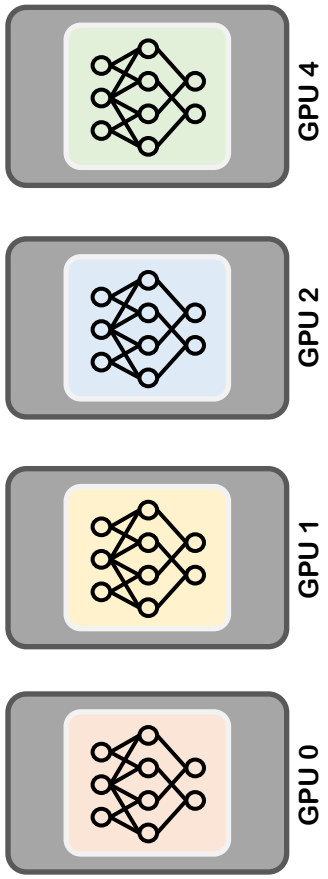




Data Parallelism

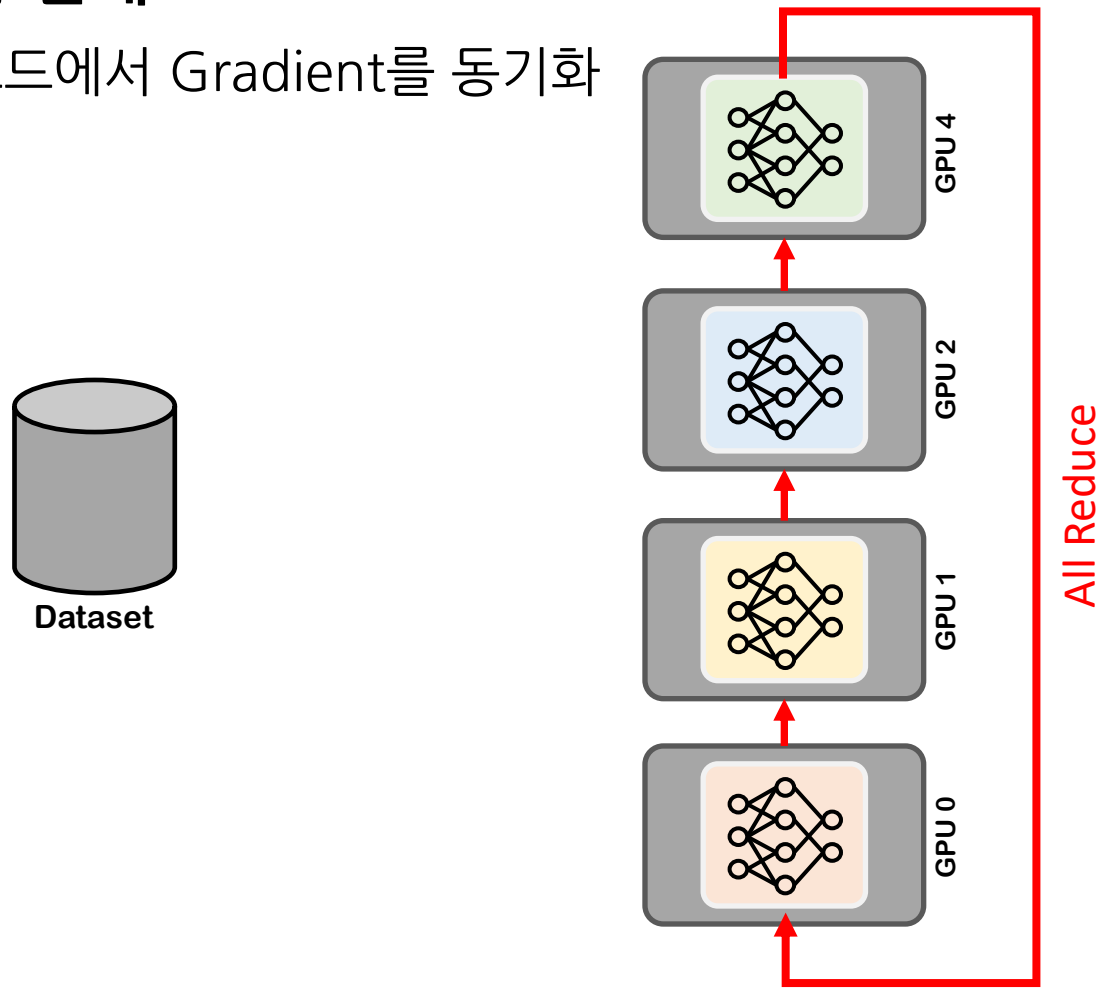
- **데이터 병렬화 기법의 학습 단계**
  - 각 모델은 서로 다른 Gradient를 갖게 됨





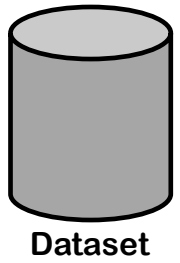
Data Parallelism

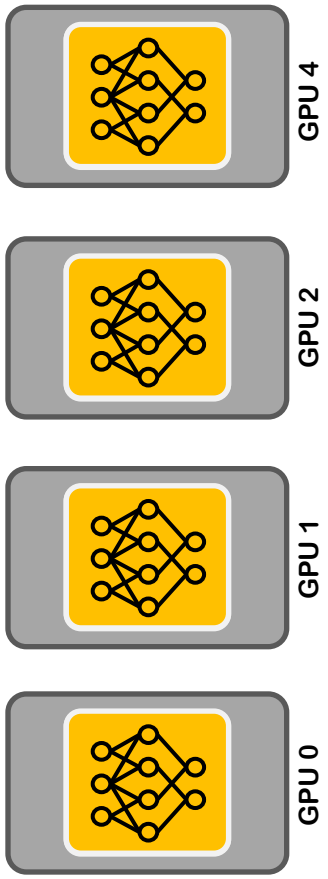
- **데이터 병렬화 기법의 학습 단계**
  - 분산 학습에 참여한 모든 노드에서 Gradient를 동기화



Data Parallelism

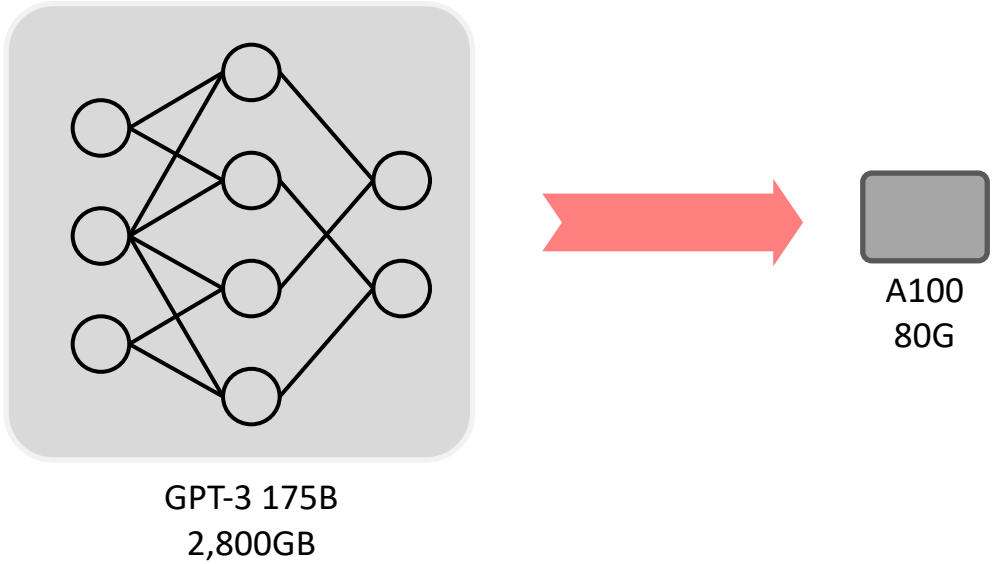
- **데이터 병렬화 기법의 학습 단계**
  - 동기화된 Gradient로 모델 업데이트





Model Parallelism

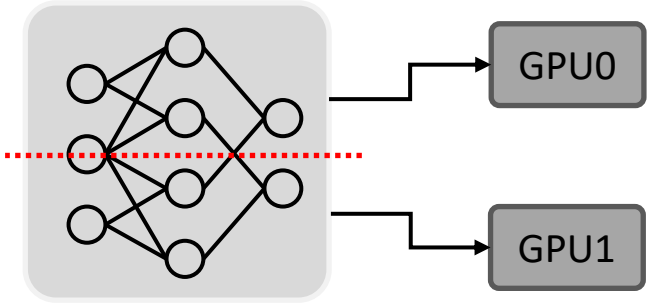
- **GPT-3 175B 학습 시 메모리 요구량 : 2.8TB**
  - Parameters : 700GB (175B * 4Bytes)
  - Gradient : 700GB
  - Optimizer : 1400GB

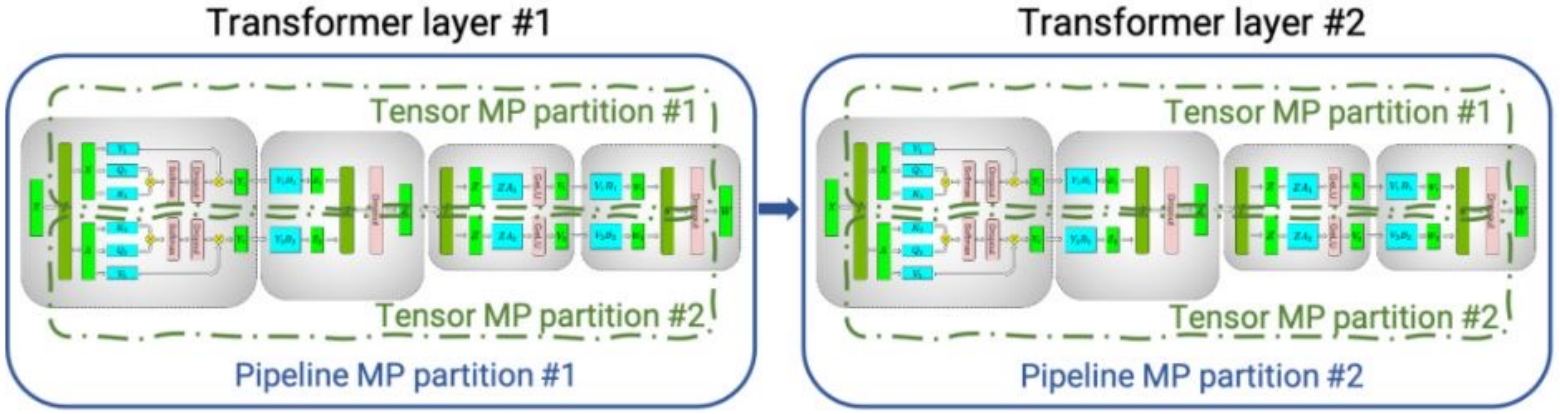


Model Parallelism

- **Intra-layer model parallelism**
  - 모델의 Tensor를 쪼개는 방식
  - **Tensor parallel**
- **Inter-layer model parallelism**
  - 모델의 layer를 기준으로 쪼개는 방법
  - 효율성을 개선한 **Pipeline parallel**이 일반적으로 쓰임

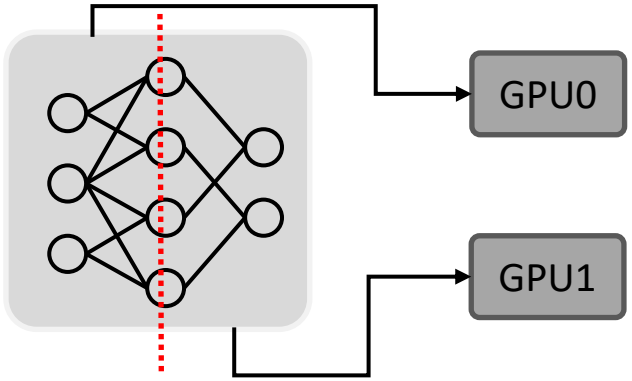
Intra-layer model parallelism





Megatron-LM model parallel

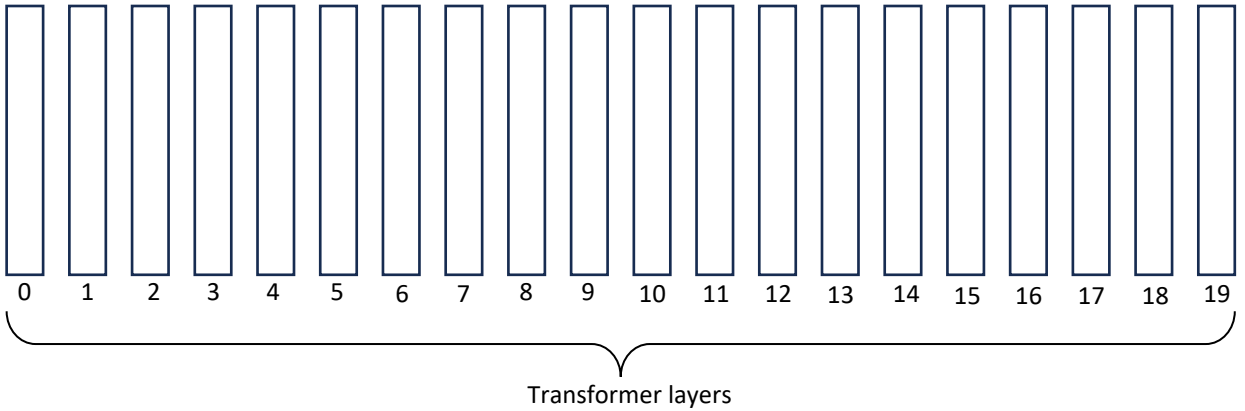
Inter-layer model parallelism



3D Parallelism

- **3가지 병렬화 기법을 동시에 적용**
  - Tensor parallel
  - Pipeline parallel
  - Data parallel

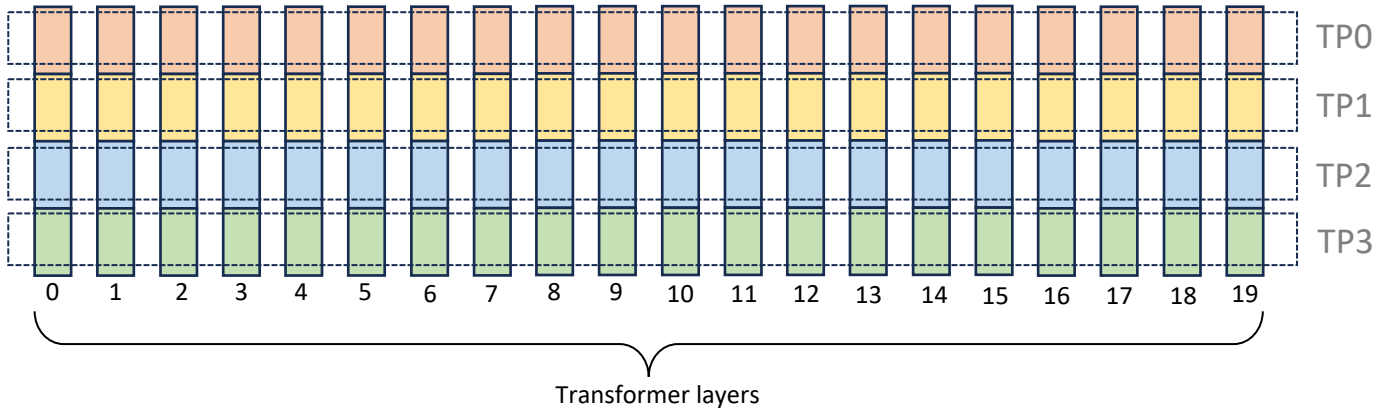
Tensor parallel : 4
Pipeline parallel : 4
Data parallel : 2



3D Parallelism

- **병렬화 기법을 동시에 적용**
  - Tensor parallel
  - Pipeline parallel
  - Data parallel

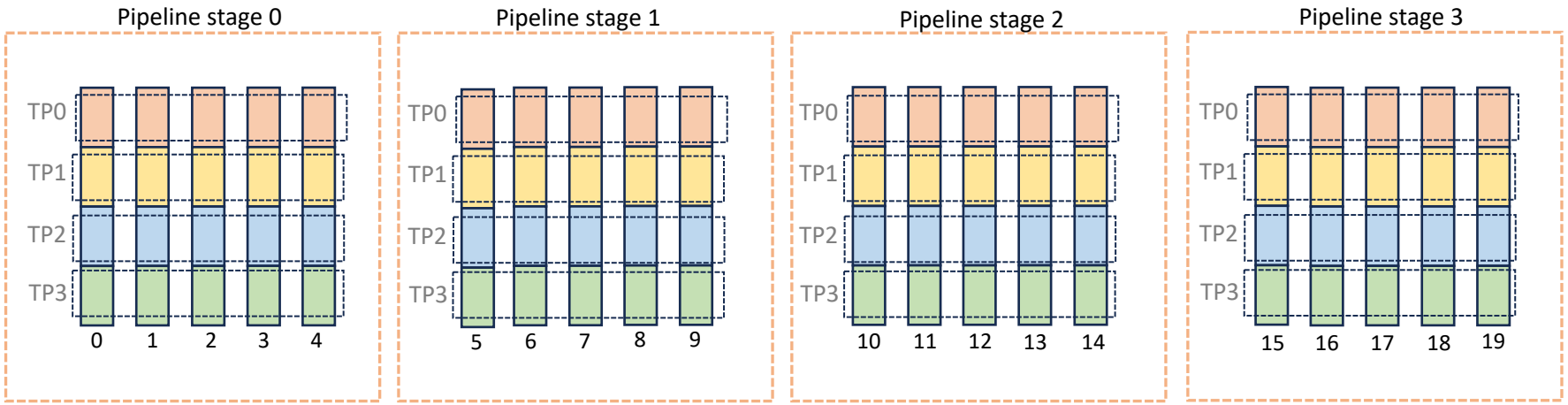
Tensor parallel : 4
Pipeline parallel : 4
Data parallel : 2



3D Parallelism

- **병렬화 기법을 동시에 적용**
  - Tensor parallel
  - Pipeline parallel
  - Data parallel

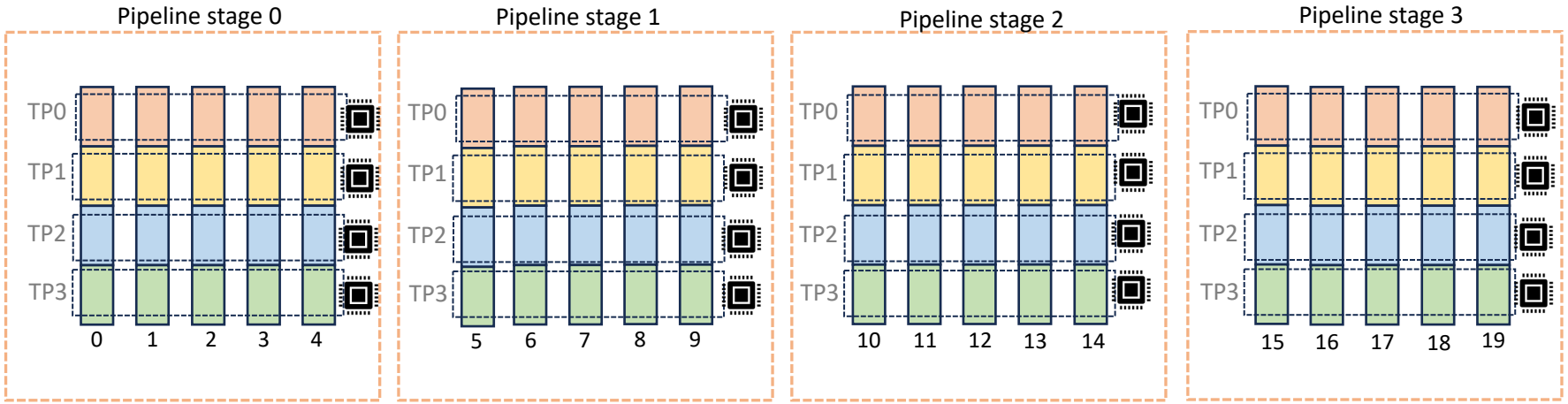
Tensor parallel : 4
Pipeline parallel : 4
Data parallel : 2



3D Parallelism

- 병렬화 기법을 동시에 적용
  - Tensor parallel
  - Pipeline parallel
  - Data parallel

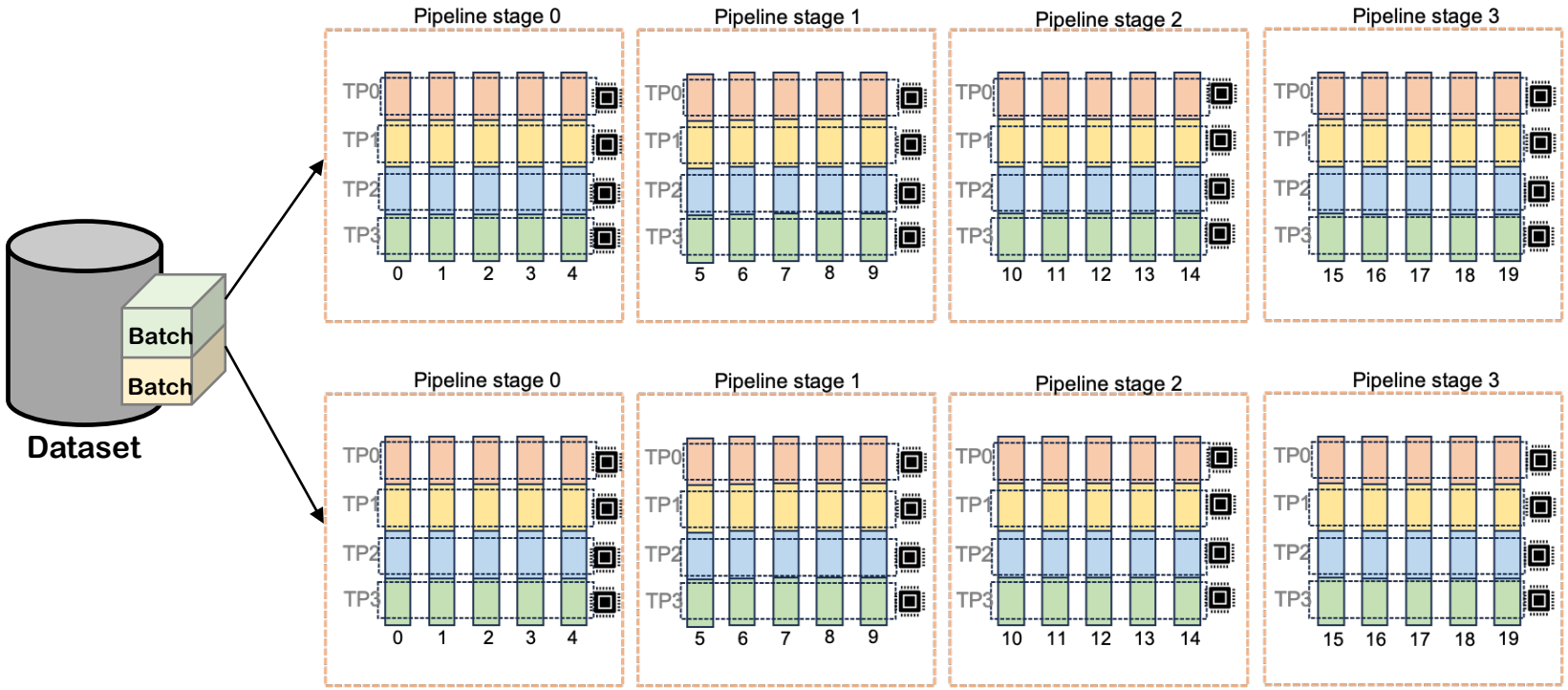
Tensor parallel : 4
Pipeline parallel : 4
Data parallel : 2



3D Parallelism

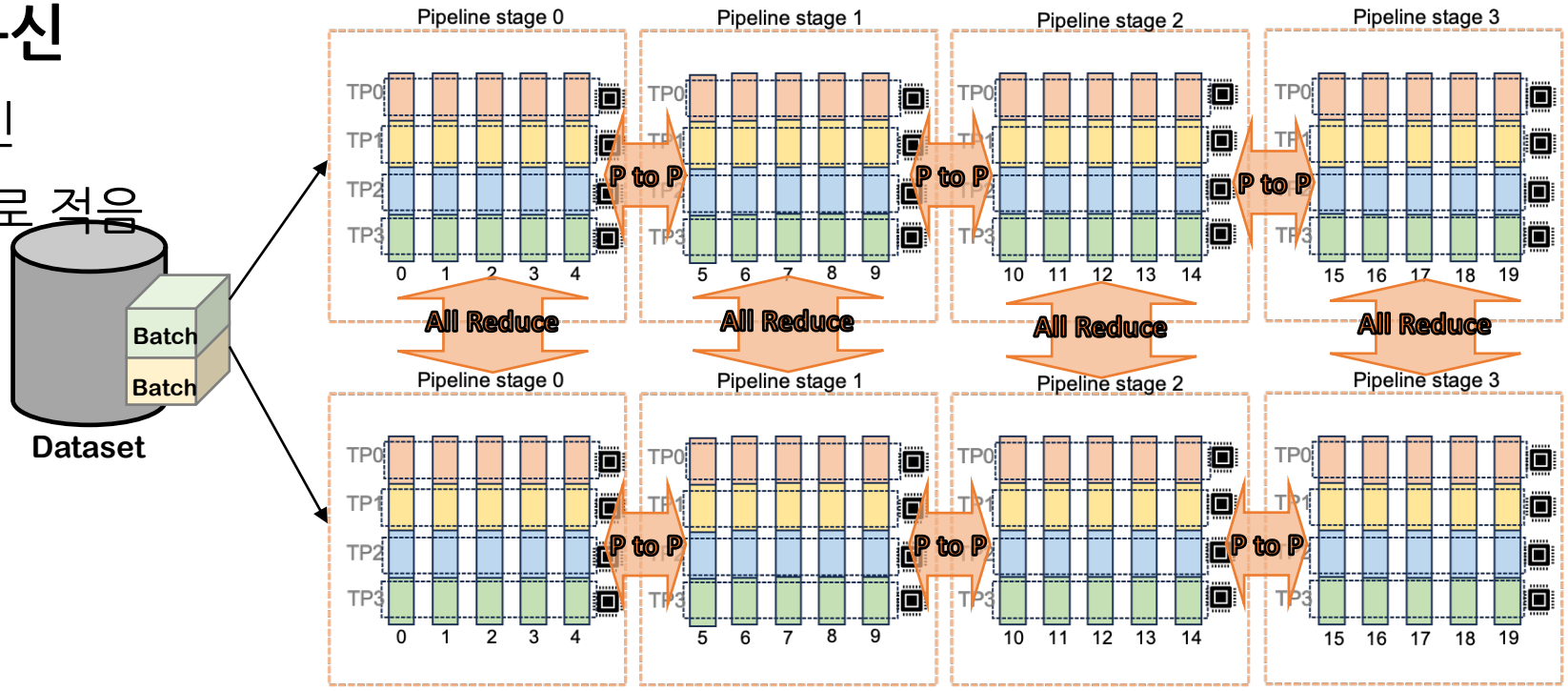
- 병렬화 기법을 동시에 적용
  - Tensor parallel
  - Pipeline parallel
  - Data parallel

Tensor parallel : 4
Pipeline parallel : 4
Data parallel : 2



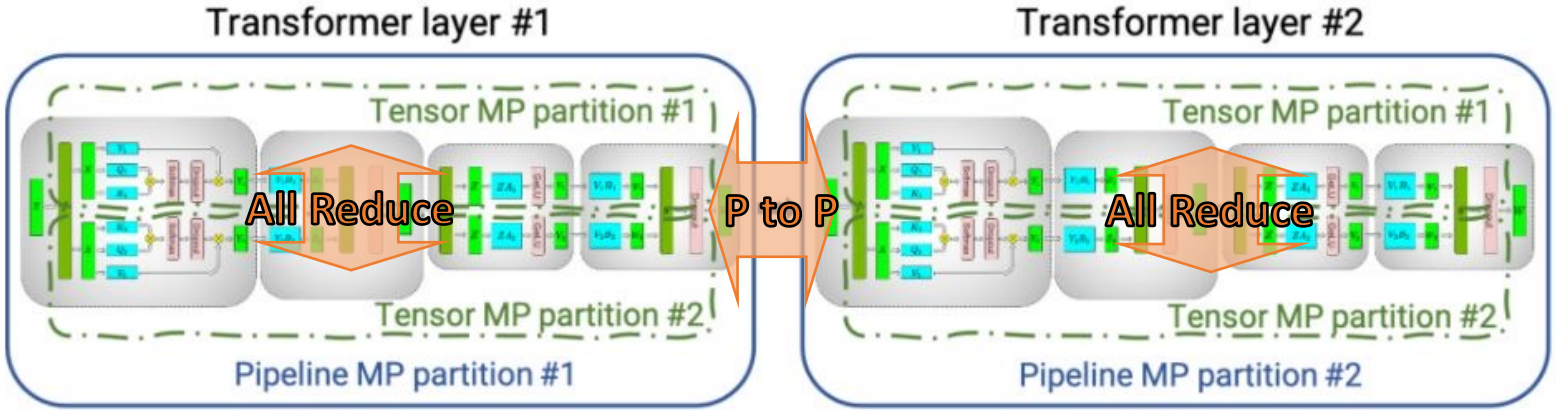
3D Parallelism

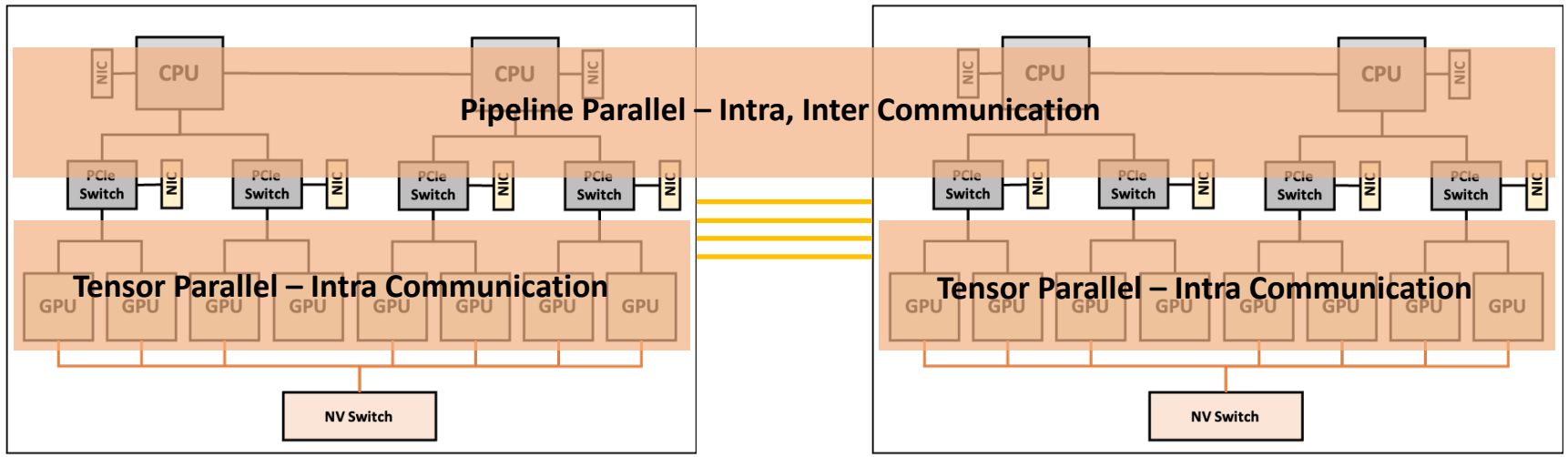
- Tensor Parallel 통신
  - All-reduce 통신
  - 통신량이 많음
- Pipeline Parallel 통신
  - Point to Point 통신
  - 통신량이 상대적으로 적음



3D Parallelism

- Tensor Parallel 통신
  - Intra communication에 배치
- Pipeline Parallel 통신
  - Intra, Inter communication 에 배치

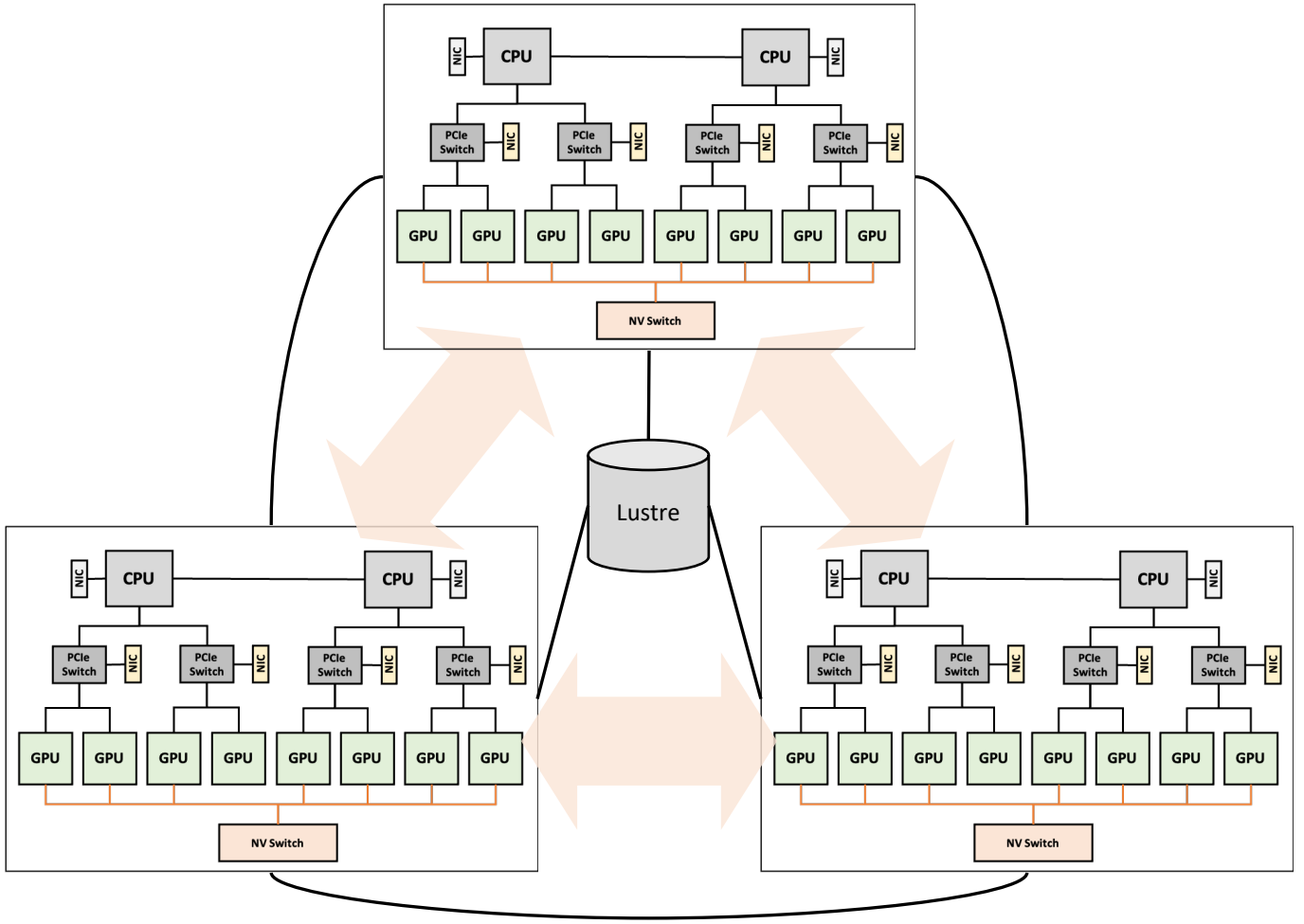




Hardware Stack

H/W Stack - 주요 고려 사항

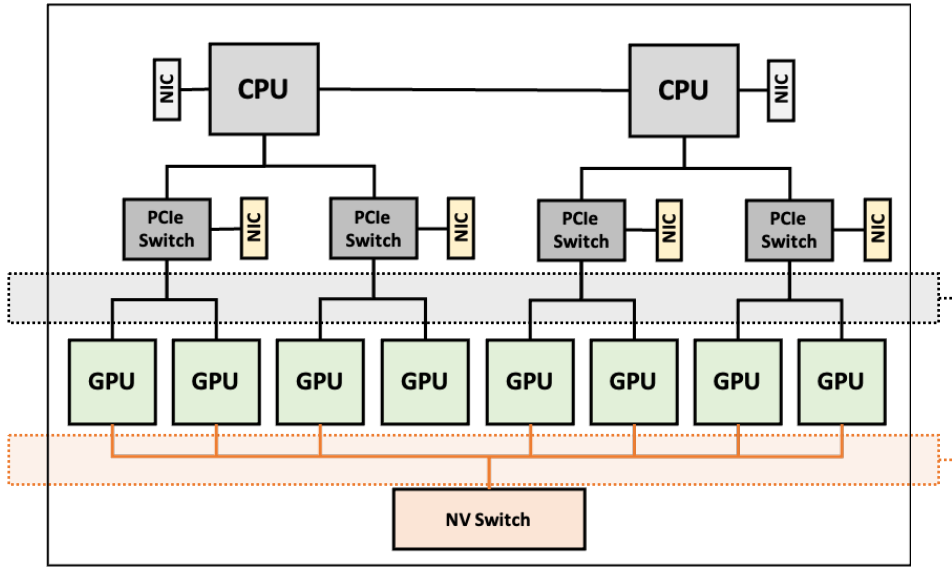
- **시스템 내부 및 시스템 간 통신의 최적화**
  - Intra-system communication
  - Inter-system communication
- **시스템의 처리 속도와 효율성 최적화**
  - GPU/CPU 성능 최적화
  - Main Memory
  - System Storage
- **데이터 전송의 최적화**
  - Storage Media(NVMe, SAS)
  - 파일 I/O 최적화 및 병렬 파일 시스템
  - 스토리지 네트워크



H/W Stack - 통신 최적화

- Intra communication
  - NVLink vs PCIe

Feature	PCIe	NVLink
Bandwidth/Latency	Up to 64 GB/s (PCIe 4.0x16)/High	Up to 900 GB/s (NVLink 4.0)/Low
Compatibility	Universally compatible	specialized GPU, InfiniBand
Cost	Low	Very High



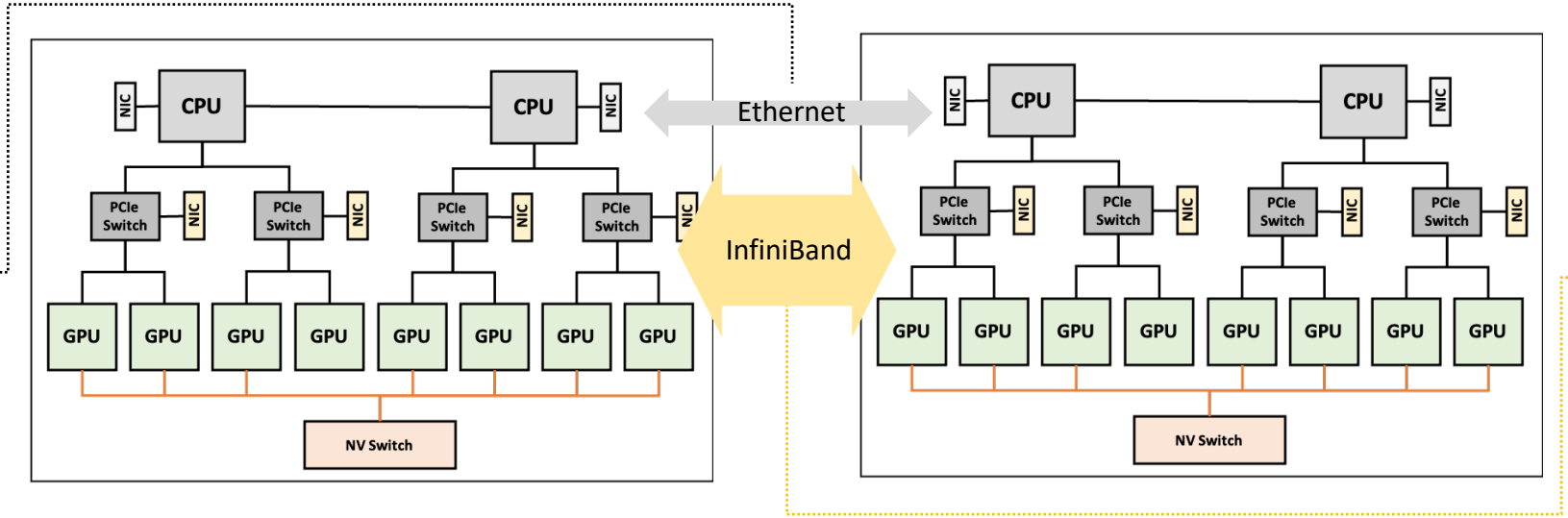
Inter connect	Bandwidths	GPU Arch
PCIe 3.0	16GB/s	Pascal, Volta
PCIe 4.0	64GB/s	Volta, Ampere
NVLink 1.0	160GB/s	Pascal
NVLink 2.0	300GB/s	Volta
NVLink 3.0	600GB/s	Ampere
NVLink 4.0	900GB/s	Hopper

H/W Stack - 통신 최적화

- Inter communication
  - InfiniBand vs Ethernet

	Ethernet	InfiniBand
Bandwidth / Latency	Up to 100Gbps / High(ms)	Up to 400Gb / Low(ns)
Protocol	TCP/IP	RDMA
Compatibility	Universally compatible	specialized high-performance computing
Cost	Low	Very High

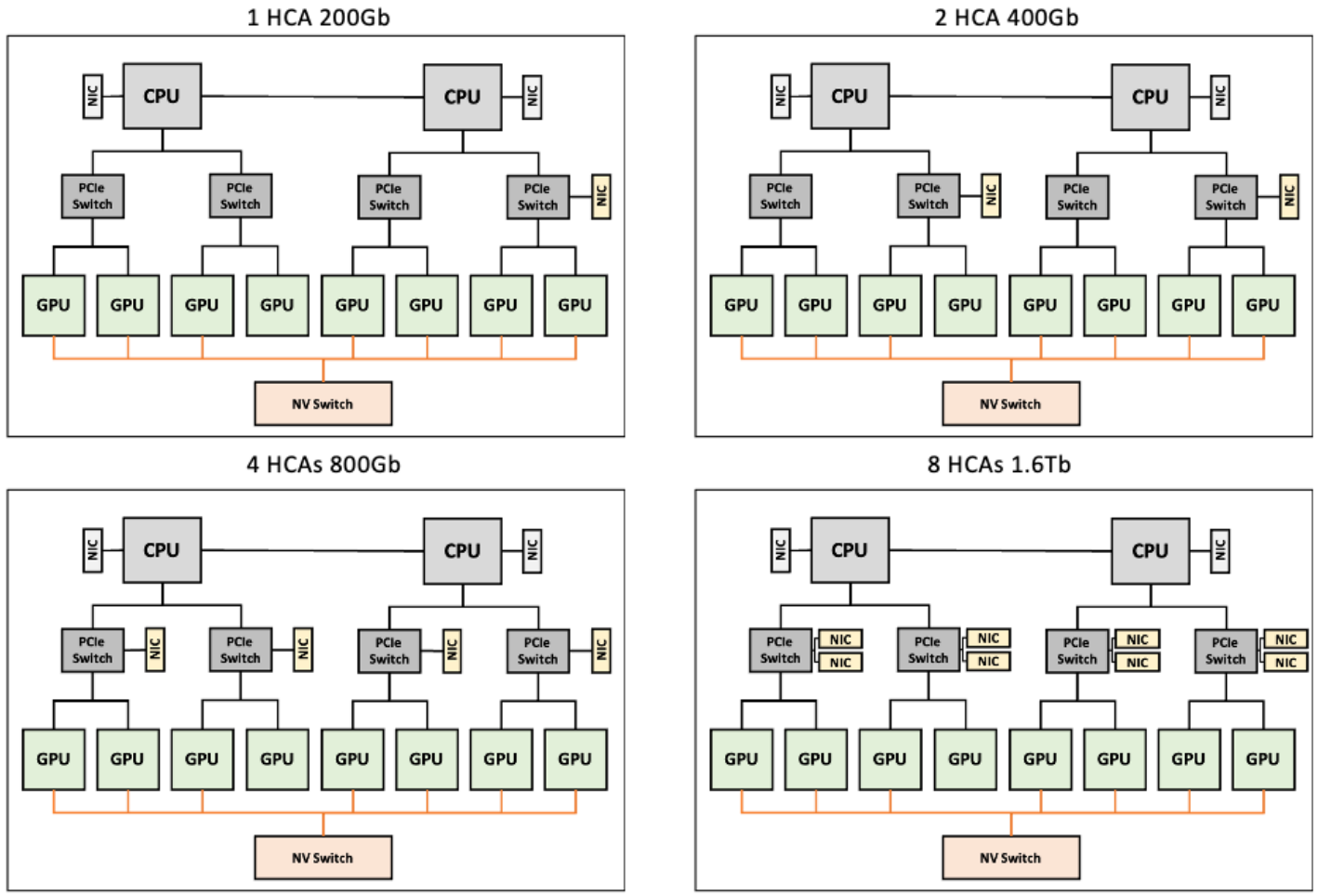
Inter connect	BandW
Ethernet 10GbE	10Gbps
Ethernet 25GbE	25Gbps
Ethernet 25GbE	50Gbps
Ethernet 100GbE	100Gbps



Inter connect	BandW
InfiniBand EDR	100Gbps
InfiniBand HDR	200Gbps
InfiniBand NDR	400Gbps
Omni-Path 100Gb	100Gbps

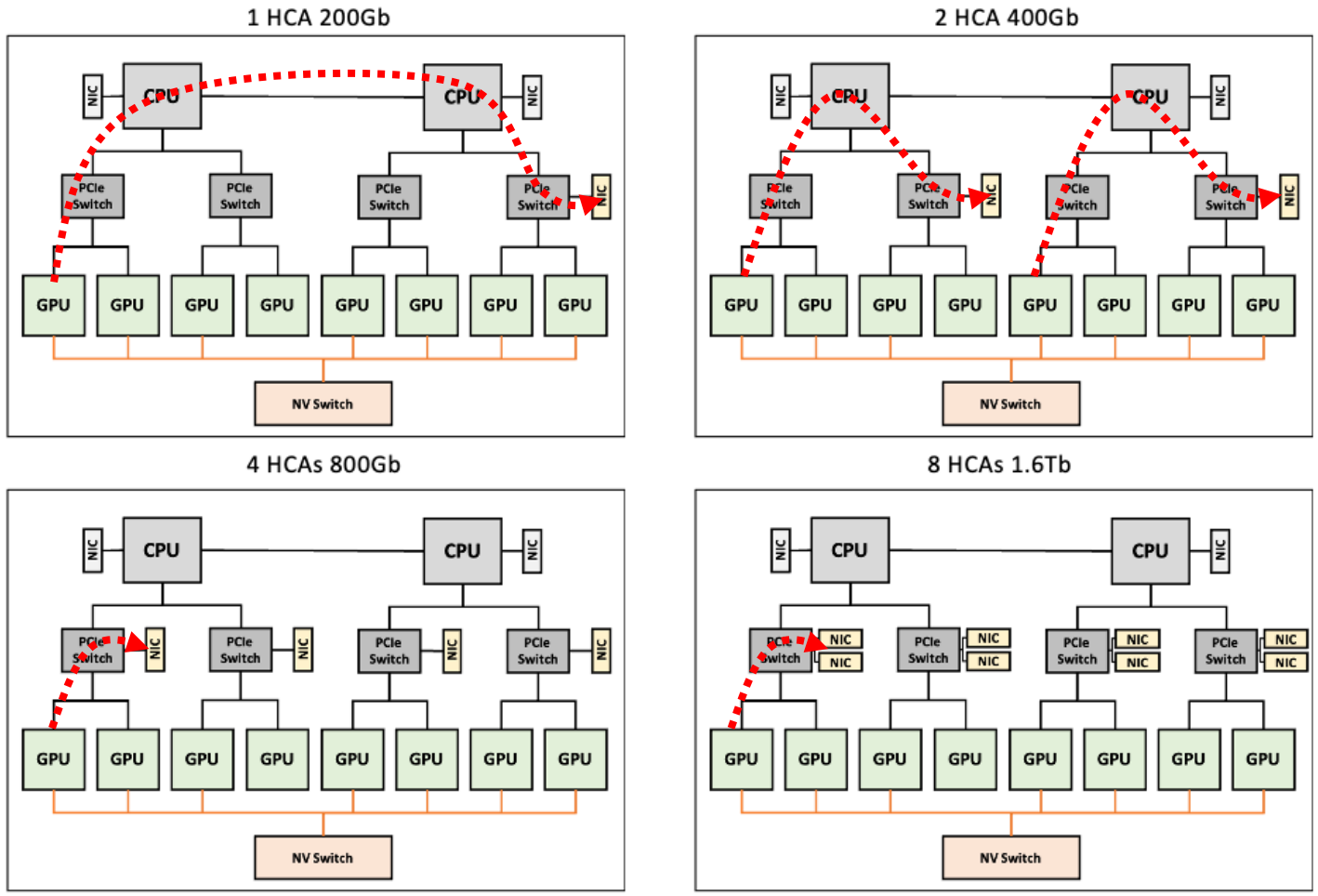
H/W Stack - 통신 최적화

- Inter communication
  - 몇개의 InfiniBand 인터페이스가 필요할까?



H/W Stack - 통신 최적화

- Inter communication
  - 몇개의 InfiniBand 인터페이스가 필요할까?
    - 1개, 2개 - Latency 증가
    - 4개, 8개 - Traffic isolation?
  - 비용 효율성 극대화가 관건

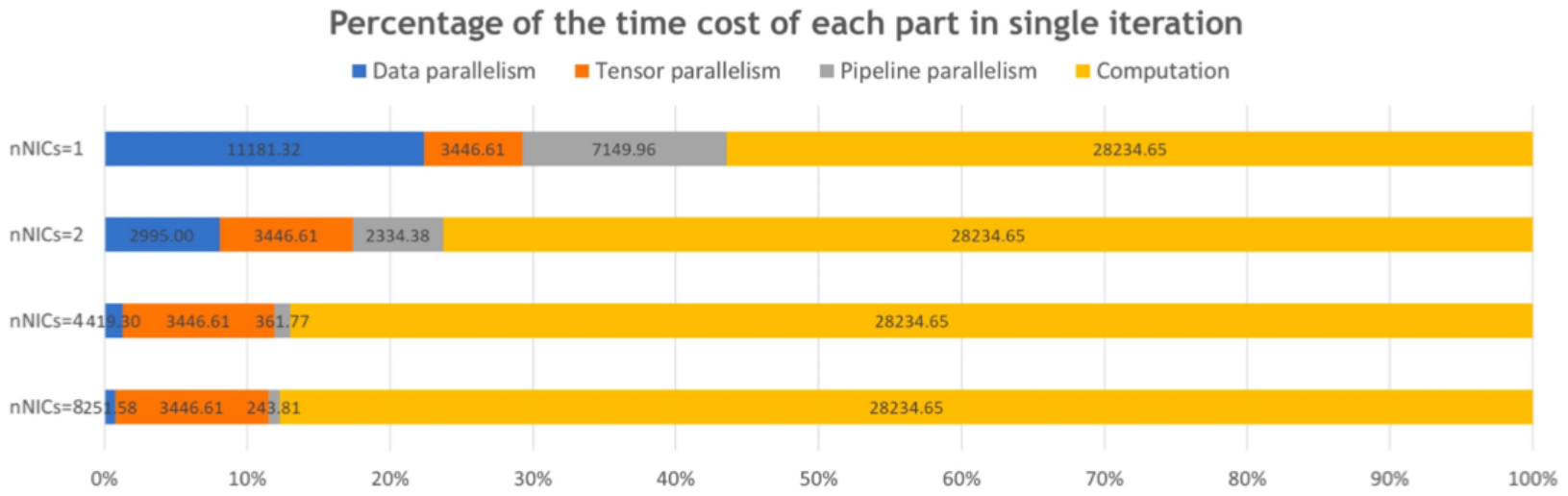


H/W Stack - 통신 최적화

- 실험을 통해 확인
  - 모델의 아키텍처와 크기에 따라 결과는 달라질 수 있음

End-to-end training for 175B model

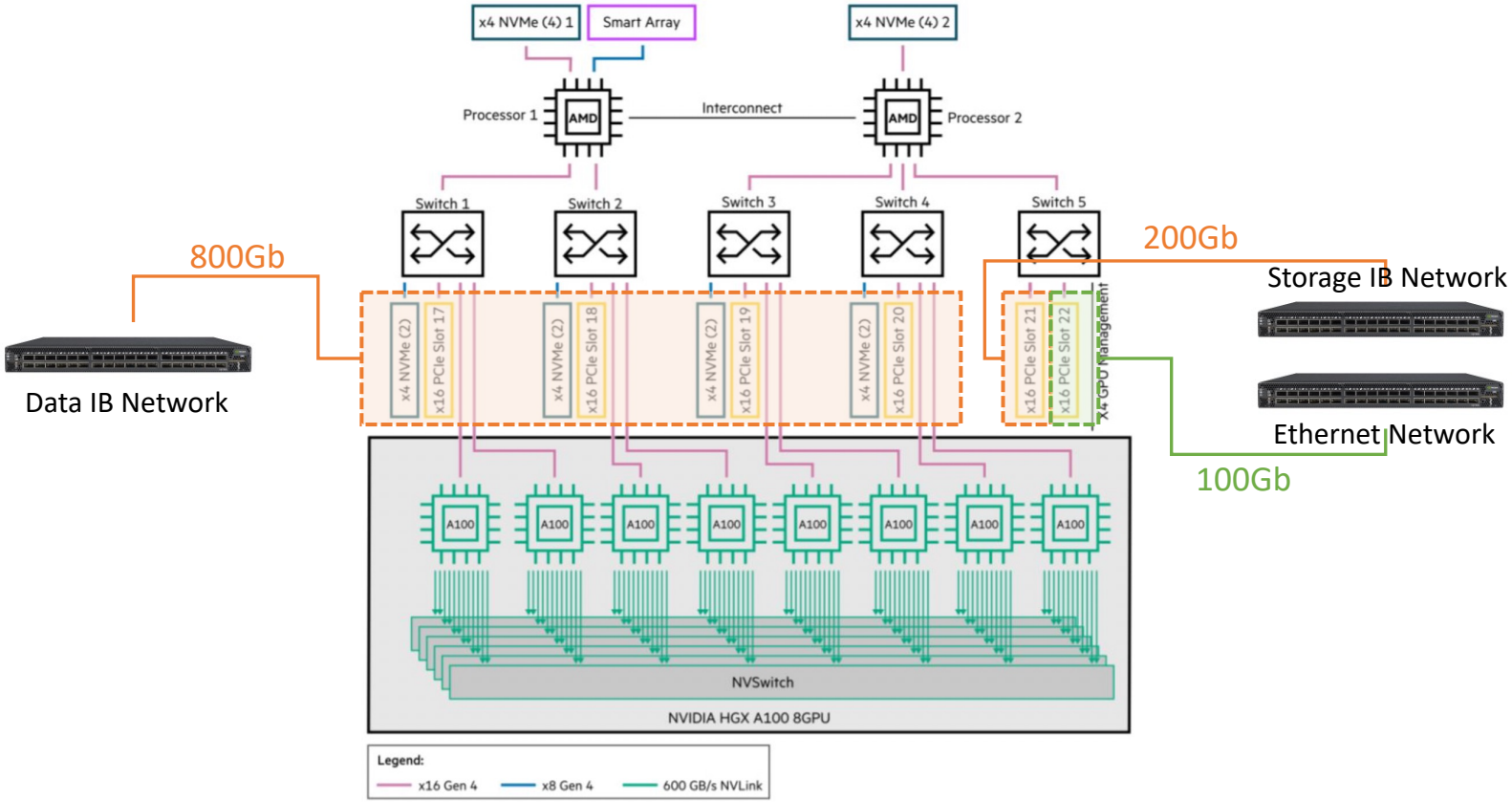
175B	8 NICs	4 NICs	2 NICs	1 NIC
Iteration time (ms)	32209	32492	37105	50368
(vs Baseline)	1.00	1.01	1.15	1.56



Hongbin, Petrick, Dive Deep into the Performance Model of GPT-3 Training on Megatron-LM GTC 2022

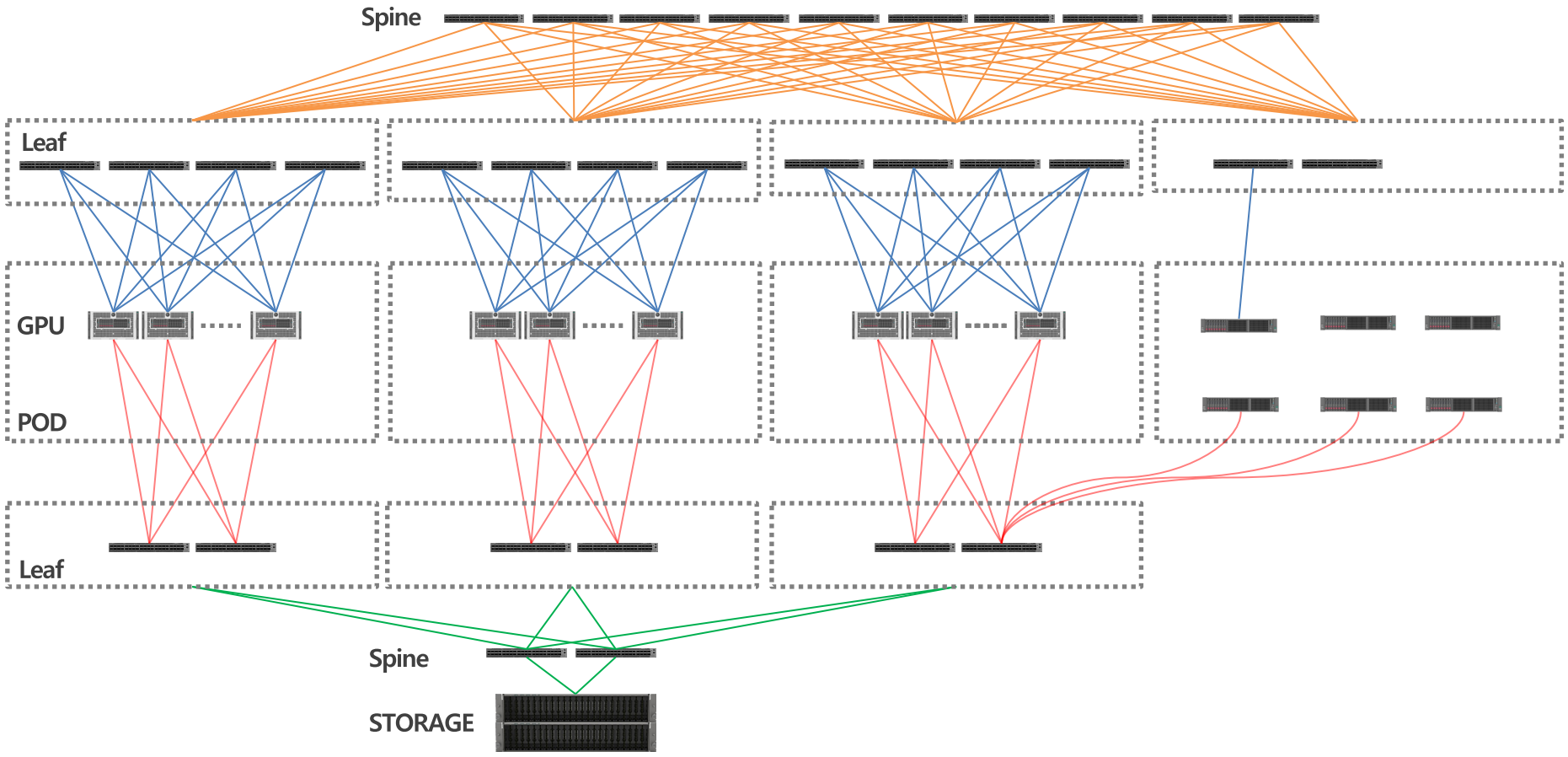
H/W Stack - 통신 최적화

- **Network 간섭 최소화**
  - GPU간 통신과 Storage 통신이 간섭 받지 않도록 분리
  - 노드 관리, Resource 제어, 모니터링 등은 Ethernet을 통해



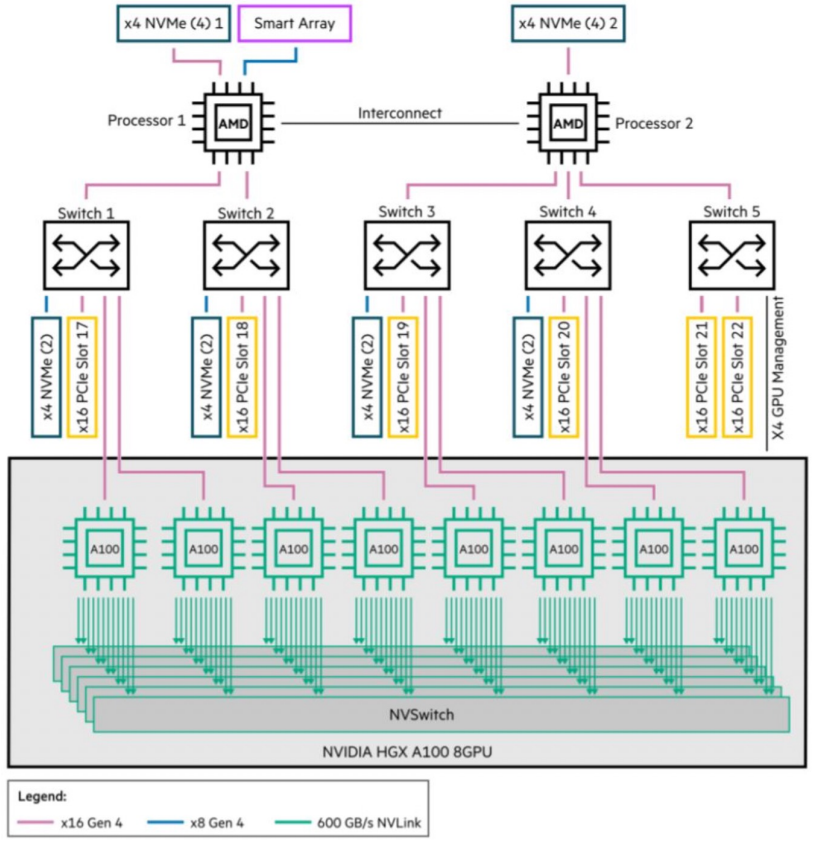
H/W Stack - 통신 최적화

- **InfiniBand Fabric Network**
  - Leaf-Spine 구조의 200G HDR InfiniBand Fabric Network
  - Over-Subscription 비율 1:1로 완벽한 Non-blocking 구성



H/W Stack - 시스템 최적화

- **HPC 시스템으로의 최적화 및 통신 성능 극대화**
  - HPC에 맞춘 시스템 튜닝 및 드라이버와 라이브러리 호환성 검증
  - Collective Communication 최적화

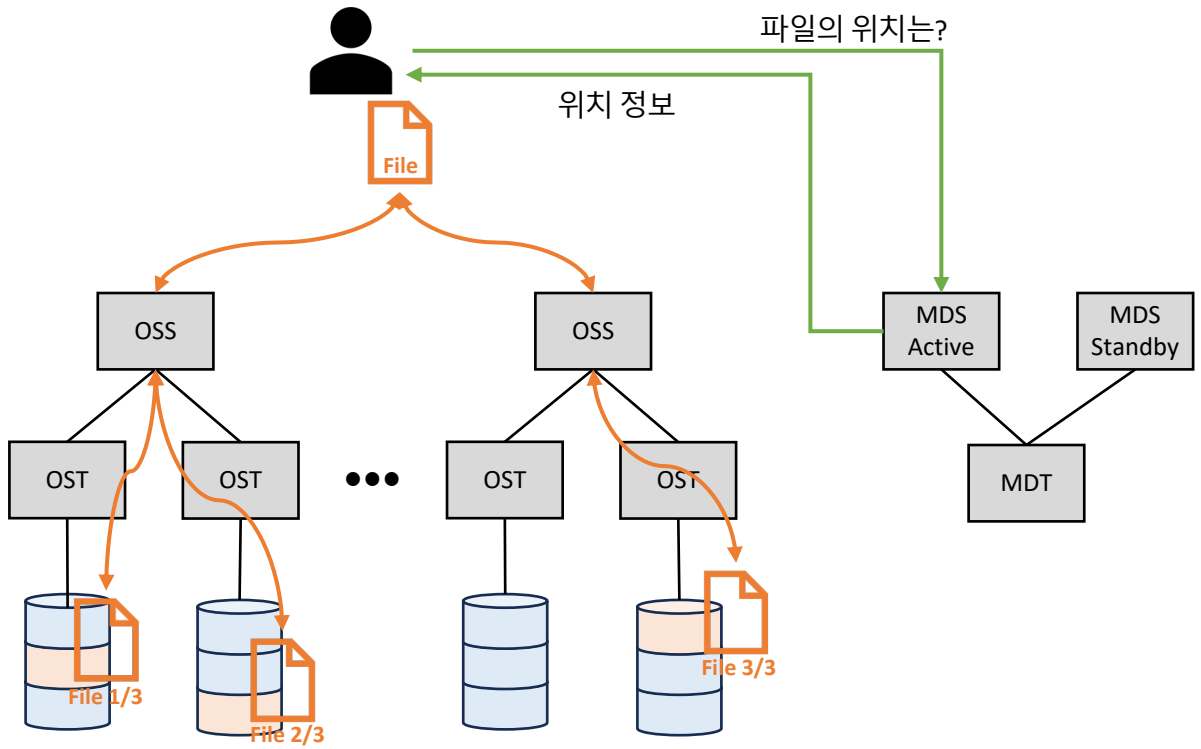


- **System 구성**
  - CPU AMD EPYC 7763 64core * 2
  - Memory : 2TB
  - Internal Storage : NVMe 35TB
- **System Tuning**
  - HPC Env Optimization (BIOS, Firmware)
  - NUMA Tuning
  - Storage Caching (Dataset, Container images)
- **GPU 및 Connectivity**
  - NVIDIA A100 80G / NVLINK 3.0 / InfiniBand
  - Driver 및 library 호환성 검증
    - OFED, MLNX_OFED, OpenMPI, SHARP, HCOLL, UCX, nvidia-driver, CUDA, NCCL
- **Collective Communication Tuning**
  - GPU Direct RDMA
  - NCCL 성능 최적화
    - NCCL_ALGO (Ring, Tree, Collnet)
    - NCCL_MIN_NCHANNELS, etc.

OPEN Compute Project®

H/W Stack - 스토리지 최적화

- **대규모 동시 I/O 에 대응**
  - 분산학습에서 일어나는 대규모 동시 읽기/쓰기에 대응
  - 병렬 분산 파일 시스템인 Lustre 채용

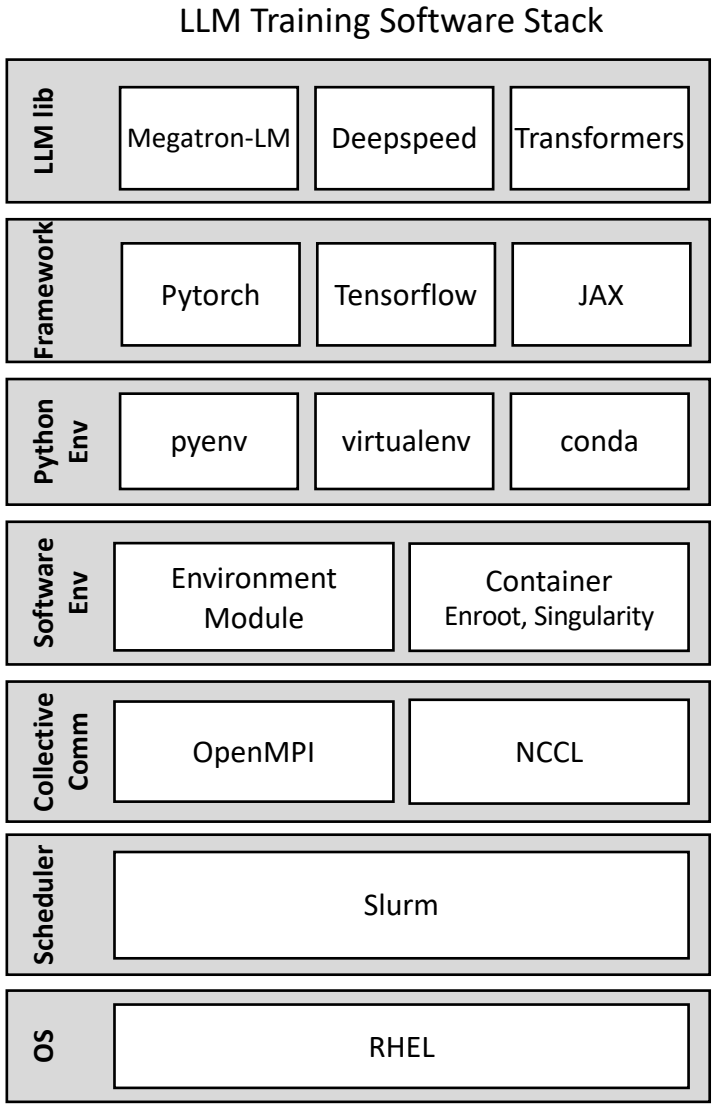


- **Lustre 병렬 분산 파일 시스템**
  - 병렬 I/O를 통해 동시 읽기/쓰기를 수행, 파일 시스템 처리 능력 향상
  - 노드 확장에 따른 처리 능력이 선형적으로 증가
- **Tiering 구조**
  - NVMe : High IOPS, TB급
  - SAS : High Throughput and High capacity, PB급
- **InfiniBand 스토리지 네트워크**
  - InfiniBand HDR 200Gb 연동
  - GPU Direct Storage를 통해 Storage와 GPU 메모리간 직접적인 데이터 접근으로 처리 능력 향상

Software Stack

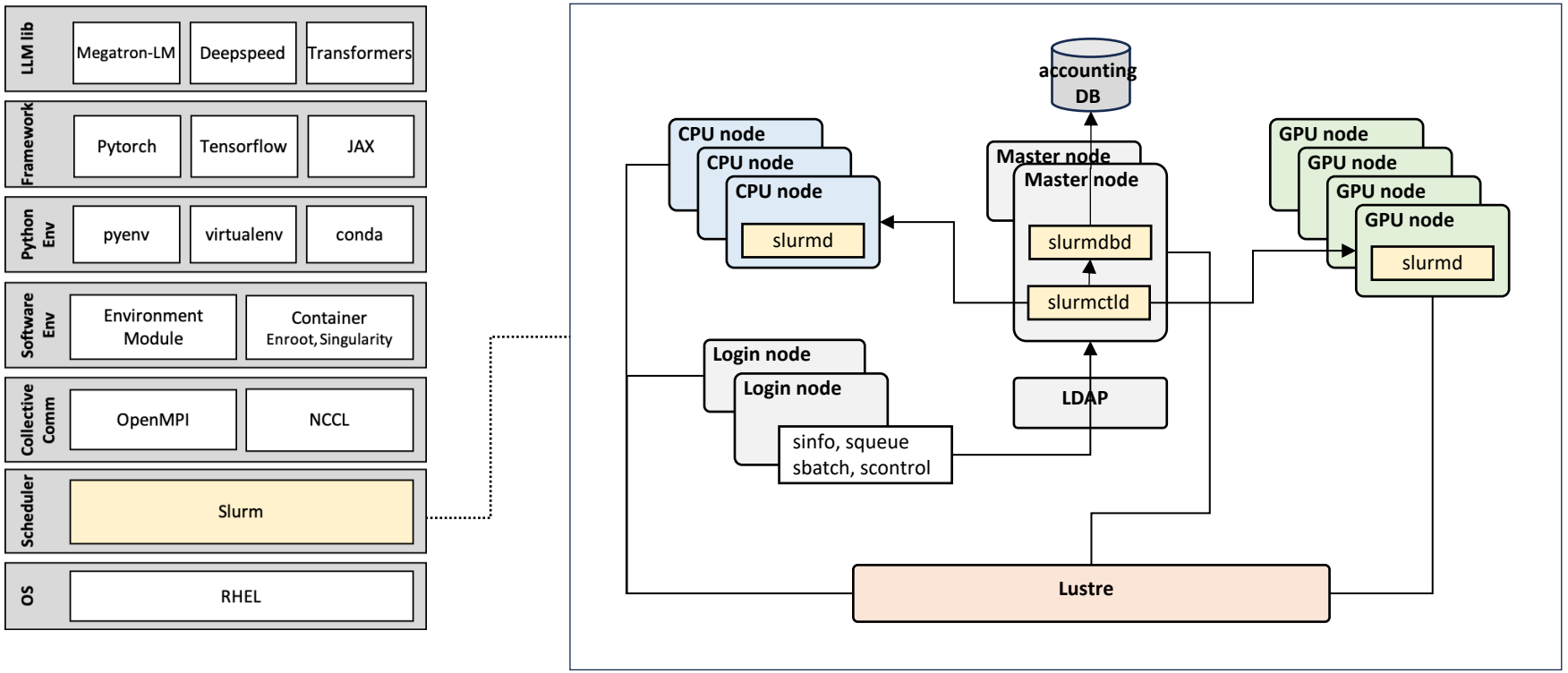
S/W Stack - 주요 고려 사항

- **Job 스케줄링 및 리소스 관리**
  - 확장성 있는 대규모 분산 학습의 실행
  - 공정한 시스템 자원의 사용
- **Software 실행 환경 최적화**
  - 효율적인 어플리케이션 및 라이브러리 관리
  - 신속하고 효율적인 실행 환경 구성
  - 개발 편의성 제공
- **고성능 Container Runtime**
  - High performance container runtime
  - Unprivileged



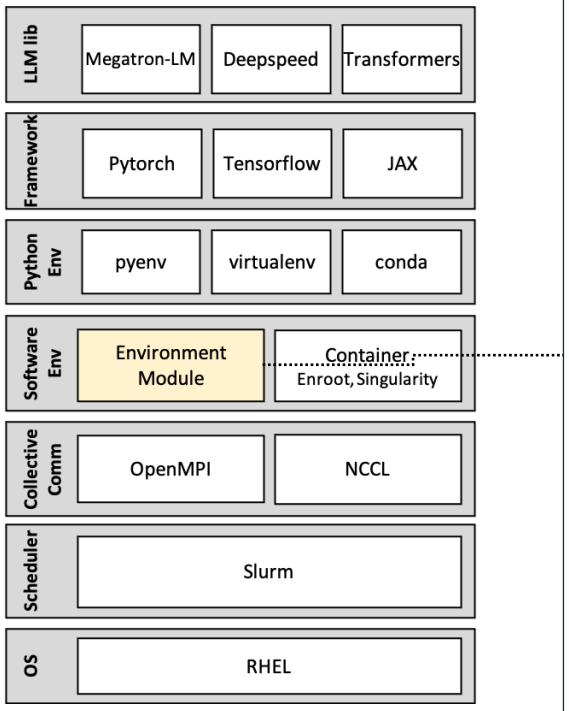
S/W Stack - Job 스케줄링 및 리소스 관리

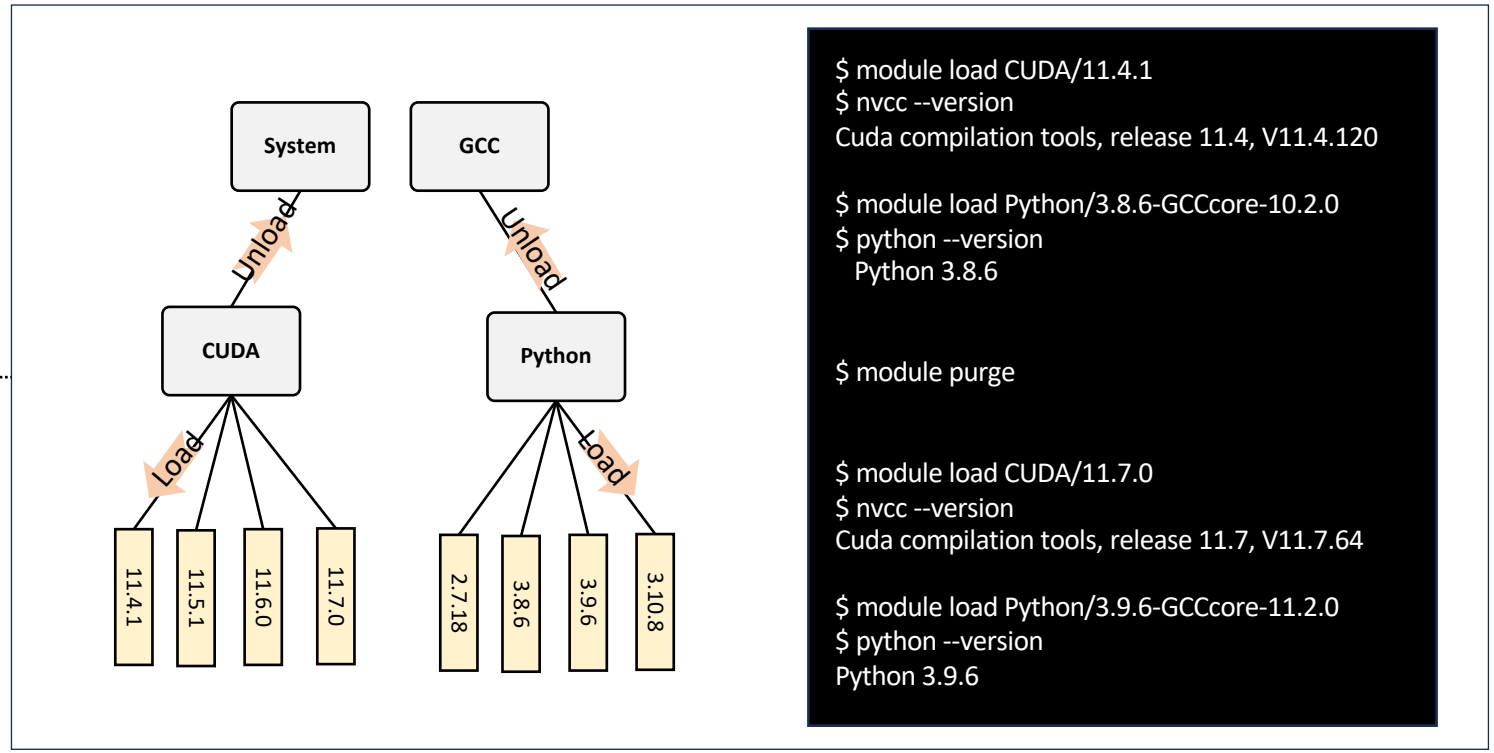
- Why Slurm?
  - 고성능 컴퓨팅(HPC)의 표준 스케줄러
  - Batch 스케줄링, Gang 스케줄링, Backfile 스케줄링과 같은 다양한 스케줄링 방식을 지원
  - 작업 우선 순위 지정 및 accounting
  - Container Orchestration과 ML Workflow에 대한 지원은 부족



S/W Stack - Software 실행 환경 최적화

- Environment Module
  - 다양한 어플리케이션과 라이브러리 버전의 관리와 전환을 간편하게 지원하는 시스템
  - 구체적으로 소프트웨어 실행에 필요한 환경변수와 라이브러리 경로등을 동적으로 변경
  - 사용자는 'module load/unload' 명령을 통해 원하는 소프트웨어를 실행하거나 동적으로 변경 가능
  - 일관된 소프트웨어 환경 유지를 위해 중앙 배포 시스템을 통해 모듈을 설치하고 공유

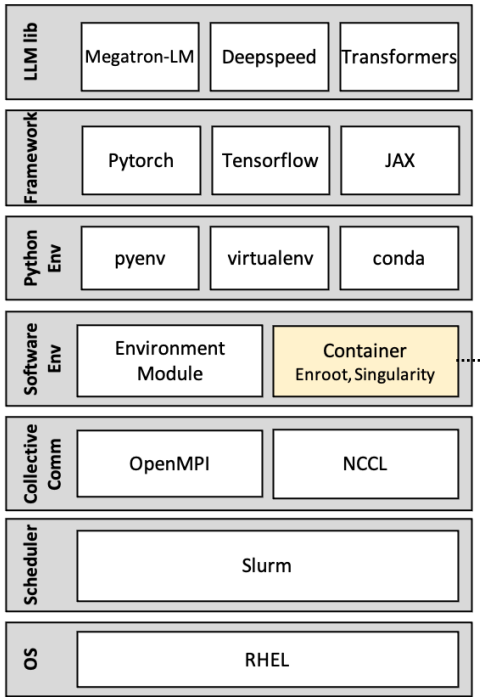


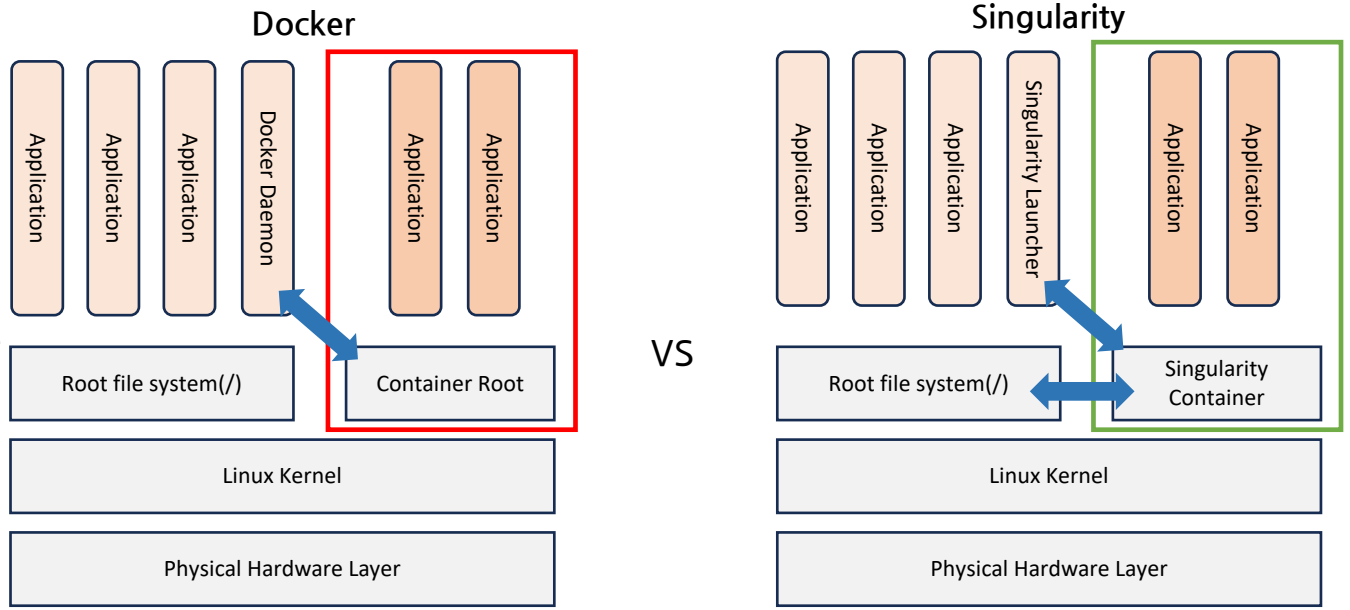



```
$ module load CUDA/11.4.1
$ nvcc --version
Cuda compilation tools, release 11.4, V11.4.120

$ module load Python/3.8.6-GCCcore-10.2.0
$ python --version
  Python 3.8.6


$ module purge


$ module load CUDA/11.7.0
$ nvcc --version
Cuda compilation tools, release 11.7, V11.7.64

$ module load Python/3.9.6-GCCcore-11.2.0
$ python --version
Python 3.9.6
```

S/W Stack - 고성능 Container Runtime

- **Enroot, Singularity**
  - User privilege 로 컨테이너 실행
  - GPU 및 네트워크 인터페이스에 직접 접근, 성능 Overhead을 제거
  - Host 파일 시스템과 통합됨에 따라 일관된 사용자 권한 관리 가능
  - Docker 이미지에 간편한 이식성 제공 및 단일 파일로 패키징 되어 관리 용이



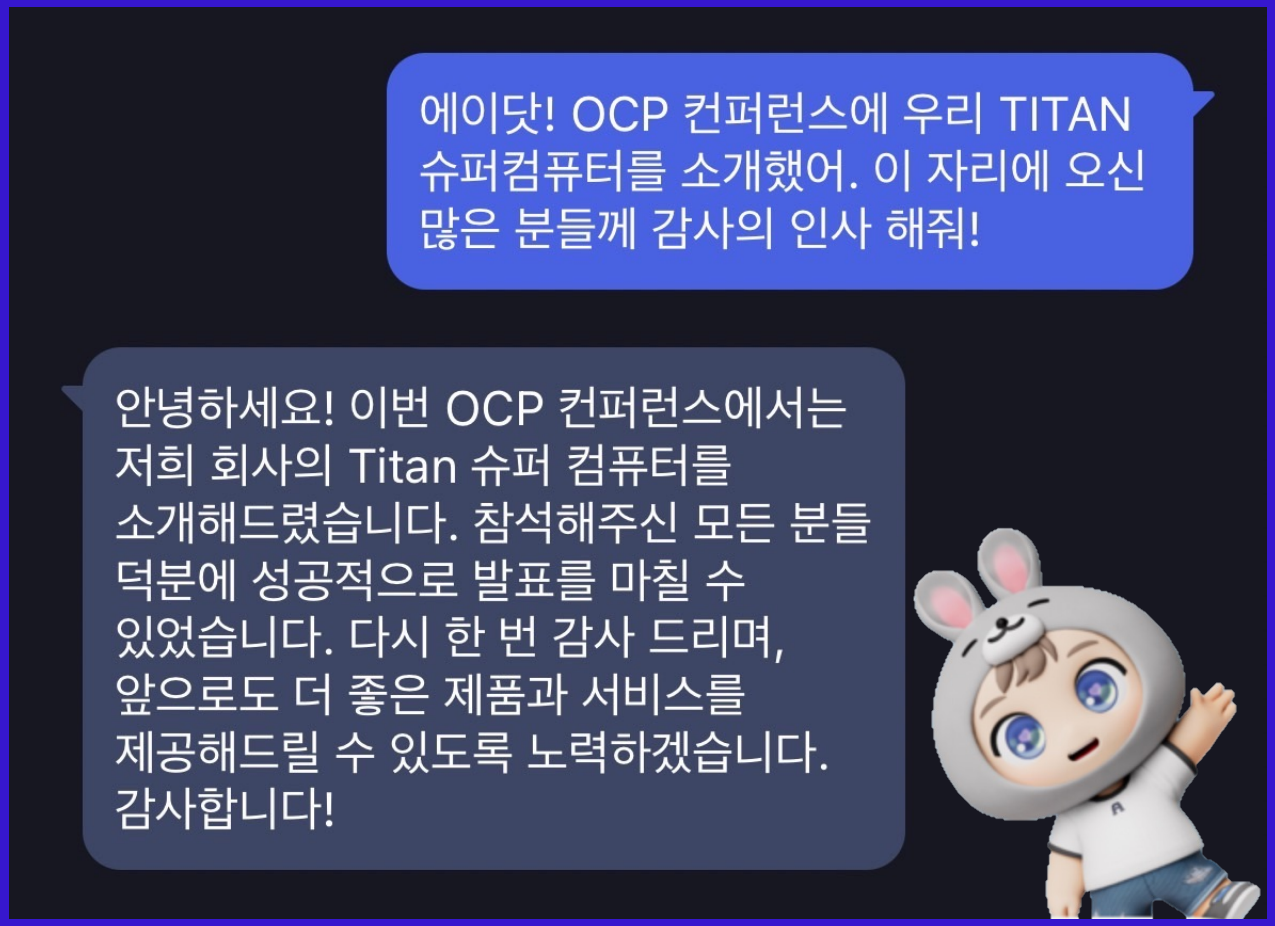


- Docker는 Docker Daemon으로 부터 컨테이너 실행
- Docker Daemon에 종속
- 독립적인 Name Space

- Singularity는 독립적으로 컨테이너 실행(single executable file)
- 특정 Daemon에 종속 되지 않음
- Host의 Name Space를 공유 가능

Guohua Li, HPC Cloud Architecture to Reduce HPC Workflow Complexity in Containerized Environments

OPEN Compute Project®



경청해 주셔서 감사합니다!