



# 대규모 GPU 클러스터에서의 Rail Optimized 및 NCCL PXN 기반 통신 최적화

이세하<sup>1</sup> 신홍일<sup>2</sup> 구건재<sup>3</sup>

<sup>1</sup> 고려대학교 빅데이터 융합학과 <sup>2</sup> 델 테크놀로지스 <sup>3</sup> 고려대학교 컴퓨터학과

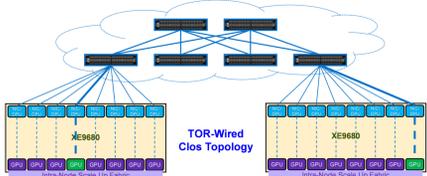
slee585@korea.ac.kr, hongil.shin@dell.com, gunjaekoo@korea.ac.kr

## 서론

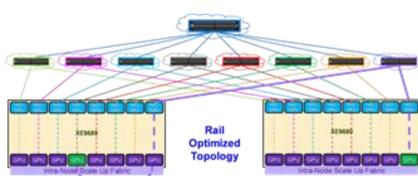
- 대규모 언어 모델(LLM) 학습에서는 GPU 간 집합 통신이 병목의 주요 원인으로 작용한다.
- NVIDIA의 PXN (Peer-to-Peer over NVLink) 기능은 CPU 경유 없이 GPU 간 직접 통신을 지원한다.
- Dell의 XE9680 서버에서는 GPU 2개당 1개의 HCA로 구성된 Rail Optimized Network를 적용해 GPU-HCA 간 최적 경로를 통해 대역폭 효율을 향상 시켰다.
- 본 연구는 NCCL PXN, CPU Affinity, NUMA 최적화를 결합하여 대규모 GPU 클러스터의 통신 성능을 개선하려고 한다

## 연구동기

- 대규모 LLM 학습에서는 수백~수천 개의 GPU가 연결되며, 네트워크 혼잡과 지연이 학습 효율을 저하시킴.
- 기존 Clos 토폴로지 (그림 1)는 확장성은 높지만, 다단계 스위칭으로 인해 통신 병목이 발생함.
- Rail Optimized Network (그림 2)는 GPU-HCA-스위치 간 직접 경로를 제공하여 경합과 지연을 줄이는 대안으로 주목받음.
- 따라서 본 연구는 Dell XE9680 서버 아키텍처 기반에서 PXN, CPU Affinity, NUMA 최적화를 통합 적용해 실질적인 성능 개선 수치와 최적화 방법론을 제시함.



(그림 1) Clos Topology 구성도 사진



(그림 2) Rail Optimized 구성도

## 기술배경 및 관련연구

### 1. NCCL PXN (Peer-to-Peer over NVLink)

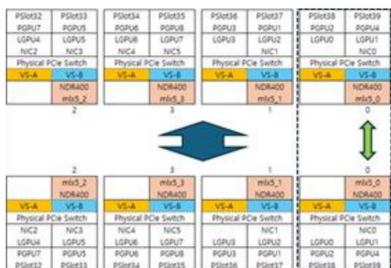
GPU 간 데이터가 CPU를 거치지 않고 NVLink 또는 PCIe를 통해 직접 전송하여 지연을 감소시킨다.

### 2. CPU Affinity & NUMA 최적화

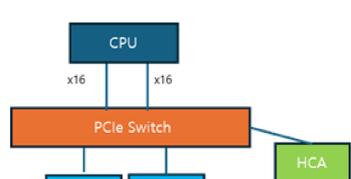
GPU-HCA 간 물리적 인접성을 고려하여 CPU 코어를 고정(taskset)함으로써 인터럽트 지연을 최소화한다. 또한 Sub-NUMA Cluster를 비활성화하여 메모리 접근 지연을 완화한다.

### 3. 제안 아키텍처 및 최적화 기법

- **NCCL PXN 활성화:** NVLink 기반 GPU 간 데이터 전송 경로를 검증하고, 특정 GPU를 선택하는 방식으로 Direct HCA 경로와 전체 GPU 경로의 성능을 비교 분석한다 (그림 3).
- **CPU Affinity 지정 및 활성화:** GPU 순서와 CPU 코어 번호를 매핑하여 스케줄링 지연을 최소화하고, taskset을 활용해 특정 코어 범위(예: 0-143)를 고정한다. NCCL의 자동 최적화 기동으로 CPU ↔ GPU ↔ NIC 간 데이터 경로에서 발생하는 불필요한 컨텍스트 스위칭을 줄이고, 통신 처리 효율을 높인다 (그림 4).
- **NUMA 최적화:** NUMA 환경에서 메모리 접근 지연은 성능 저하의 주요 원인 중 하나이다. BIOS에서 Sub-NUMA Cluster 기능을 비활성화하고, GPU와 물리적으로 가까운 CPU 코어를 고정하여 데이터 경로의 지연을 최소화한다.



(그림 3) NCCL PXN 활성화 구성도



(그림 4) CPU ↔ GPU ↔ NIC 경로 관계

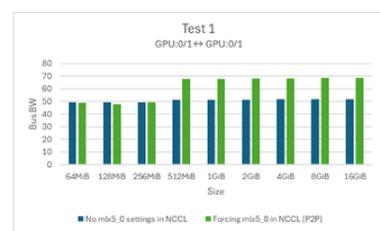
## 실험 환경 및 결과

### 실험환경

- 서버: Dell XE9680 ×2 (서버당 8×NVIDIA H100 GPU, 4×ConnectX-7 HCA, 2×Intel Xeon CPU)
- 네트워크: 400 GbE 스위치, Rail Optimized Network
- 소프트웨어: Ubuntu 22.04 LTS, CUDA 12.8, NVIDIA Driver 570.xx, NCCL 2.26+
- 측정 지표: NCCL bus bandwidth (GB/s)
- 데이터 크기: 64 MiB ~ 16 GiB

### Test 1

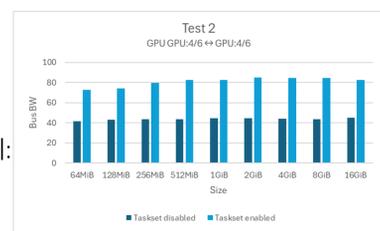
- 구성: 8×GPU - 4×HCA 환경에서 Rail Optimized 배선 적용, Direct HCA 경로(mx5\_0) 지정.
- 실험 방법: 강제로 mx5\_0이라는 Direct HCA 경로 사용하여 성능 향상되었는지 확인. (NCCL\_IB\_HCA=mx5\_0)
- 결과: 지정하기 전, 평균 bus bandwidth은 50.77 GB/s로 확인됐지만, 선택된 GPU와 가까운 HCA를 강제로 지정하여 구성했을 때, 61.77 GB/s로 약 **21.7%**의 성능 향상을 확인 (표 1).



<표 1> Forcing mx5\_0 (P2P) 사용 전 후 Bandwidth 비교

### Test 2

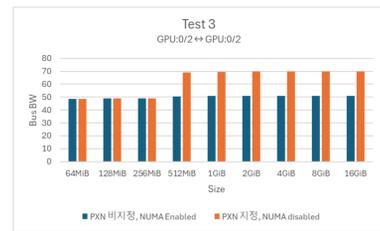
- 구성: PXN ON + Rail Optimized 환경에서 CPU Affinity 적용 전후 비교.
- 실험 방법: taskset 명령으로 특정 CPU 코어 범위(예: taskset -c 1,145)를 고정하여 GPU-CPU 매핑 수행.
- 결과: 미적용 시 43.75 GB/s → 적용 후 80.66 GB/s로 약 **47 %** 성능 향상 (표 2).



<표 2> CPU Affinity 사용 전, 후 Bandwidth 비교

### Test 3

- 구성: PXN ON + Rail Optimized + BIOS 최적화 설정(Sub-NUMA Cluster Disabled) + CPU Affinity 적용.
- 실험 방법: Sub-NUMA Cluster 설정을 기본으로 한 값과, disabled 설정한 값을 비교.
- 결과: BIOS 최적화 전 및 PXN 경로 미지정은 약 50.05 GB/s, BIOS 최적화 및 PXN 경로 지정한 값은 약 62.52 GB/s로 약 **24.8%**의 성능 향상 (표 3).



<표 3> BIOS 최적화 전, 후 Bandwidth 비교

## 결론 및 향후 연구

### 결론

- 본 연구에서는 Dell XE9680 서버 기반 환경에서 PXN 지정, CPU Affinity 적용, NUMA 최적화를 결합해 NCCL 통신 성능을 분석하였다.
- Direct HCA 경로 지정 시 약 20 %, CPU Affinity 적용 시 47 %, NUMA 최적화 시 약 **25 %**의 성능 향상을 확인하였다.
- 이는 GPU-NIC 매핑과 CPU 코어 바인딩이 고대역폭 GPU 통신 성능의 핵심 요인임을 입증한다.

### 향후 연구

- PXN, NVLink, SHARP 등 최신 네트워크 기능의 변화가 Collective 연산 구조에 미치는 영향을 정량적으로 분석해볼 것이다.
- 현재 수동으로 수행되는 PXN 경로 지정 및 CPU Affinity 설정을 자동화하는 Topology-aware Scheduler를 개발도 염두 해 두고 있다.
- 단일 노드 실험을 다중 노드로 확장하여 멀티 서버 환경(400 GbE 기반)에서의 확장성과 안정성을 검증할 예정이다.