Collective Communication Software Optimization for Heterogeneous AI Accelerators and CXL-Enabled Systems

2026-08-19

Hooyoung Ahn

Principal Researcher
Networked AI Computing Research Section
AIDC Research Division

ETRI

Contents

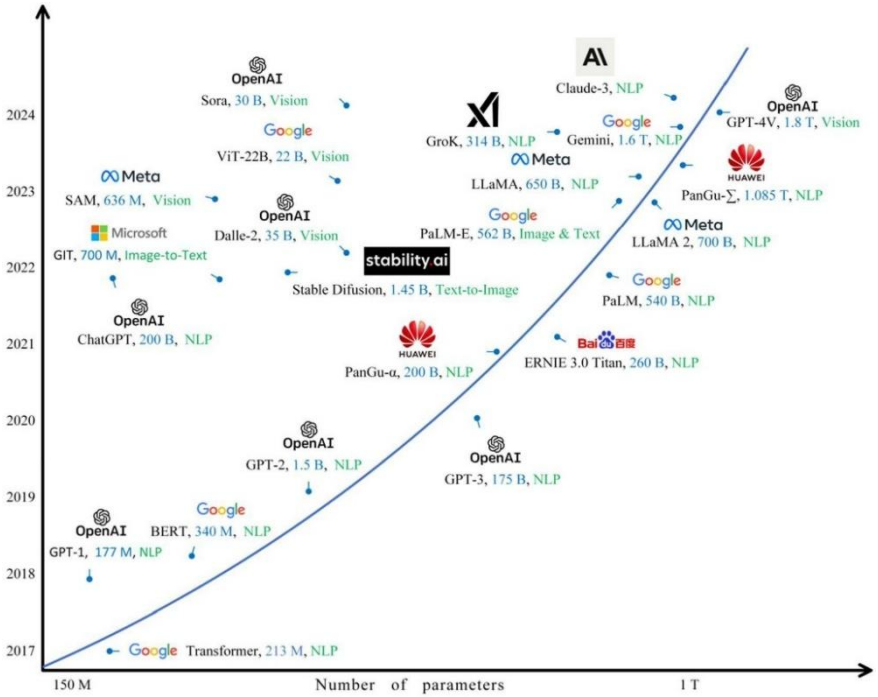
Contents

I. Research Project Overview

	Description
Project Title	Collective Communication SW Optimization for AI Accelerators (National research project funded by the Korean Government)
Organization	- ETRI (Project Leader) ETRI - HyperAccel (Korean NPU company) - MangoBoost (Korean DPU company) HYPER ACCEL MANGOBOOST
Period	5 year (From April 2026 through 2030)
Goal	To develop a vendor-neutral collective communication library, called KCCL that supports K-AI accelerators as well as NPUs, PIMs, DPUs and other AI accelerators

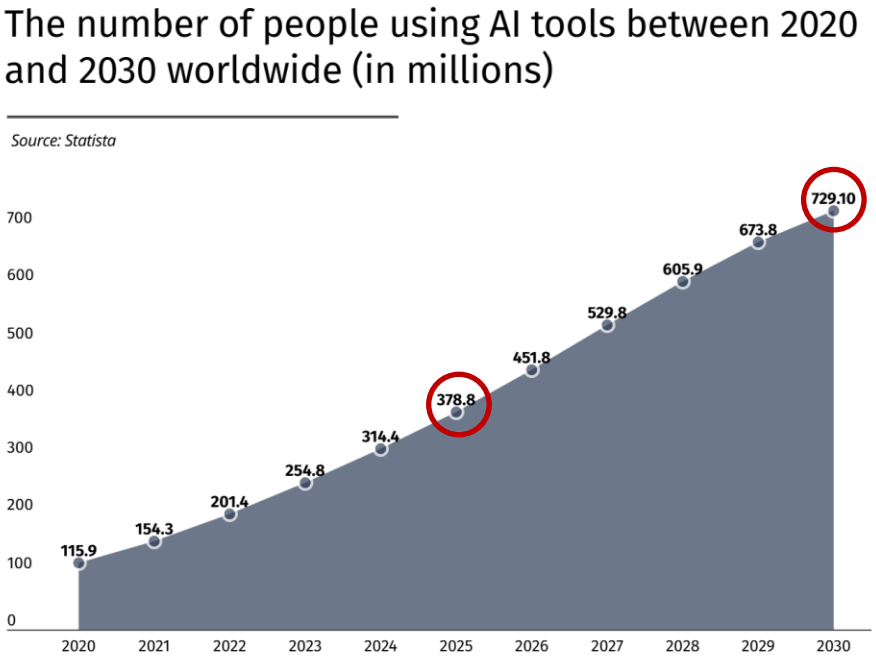
II. Background – Growth of AI Models and Users

AI models and AI users are growing explosively.
So we need much larger cloud systems built on AI accelerators.



* An overview of large AI models and their applications(Visual Intelligence '24)

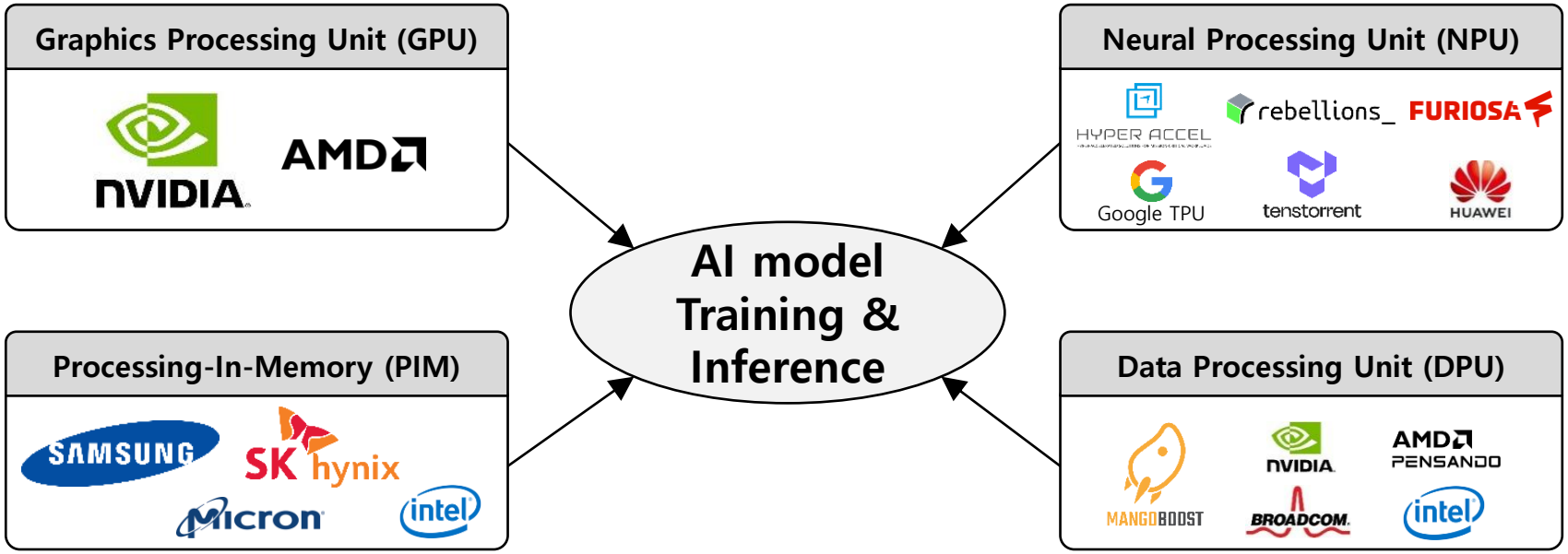
Growth trend in AI model size



Growth trend in AI users

II. Background – AI Accelerators

AI cloud systems are moving from GPU-centric designs to integrated, heterogeneous AI accelerator systems.

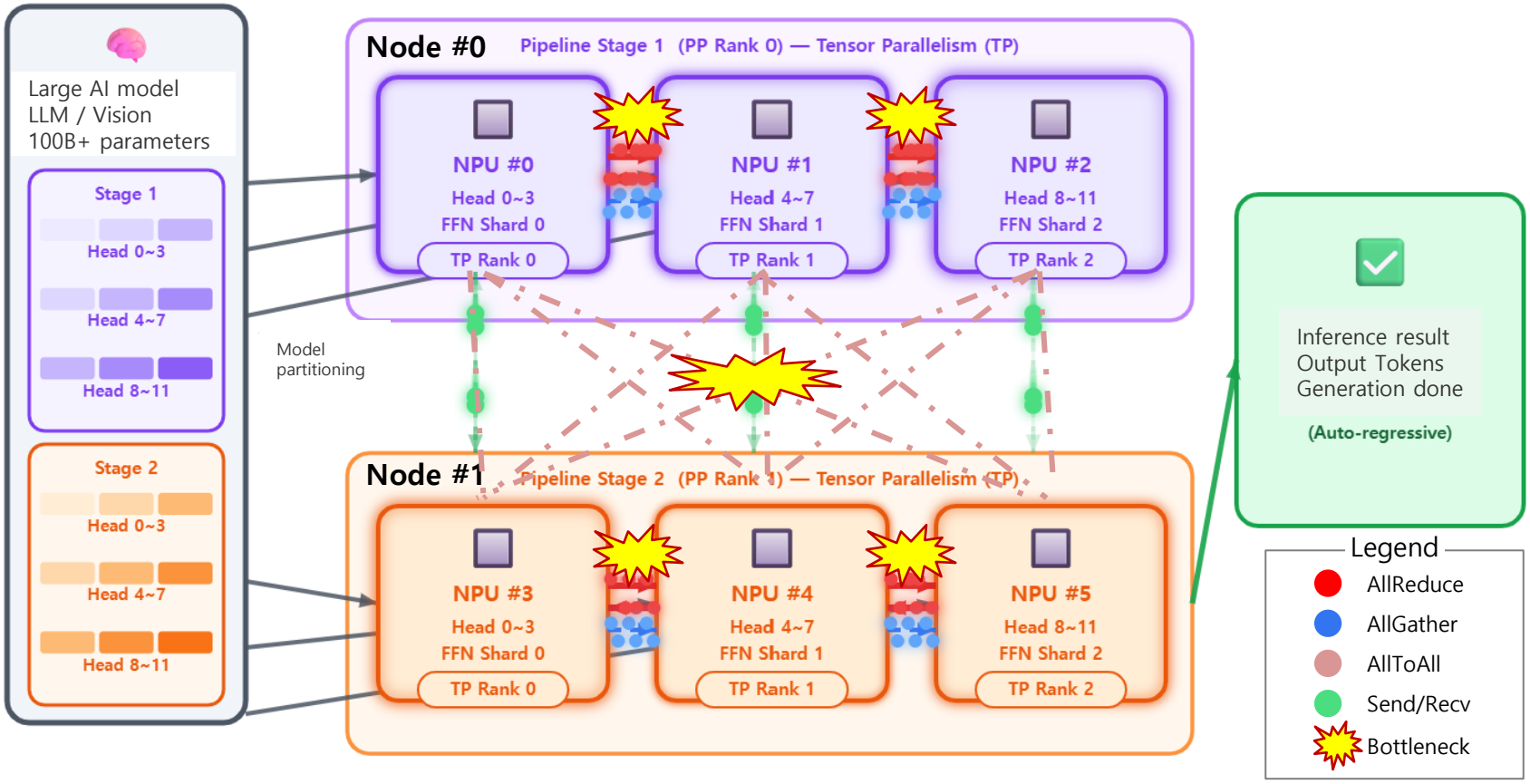


The Korean government and industry are joining forces to build globally competitive next-generation AI accelerators — Korea-developed NPUs, PIMs, and DPUs.

* AI accelerators are semiconductors designed for high-speed, high-efficiency AI training and inference.

II. Background – Hyperscale AI Computing

The hyperscale AI model runs across computing nodes and accelerators using TP/PP/EP. It is partitioned once, and then the nodes frequently communicate to synchronize data.



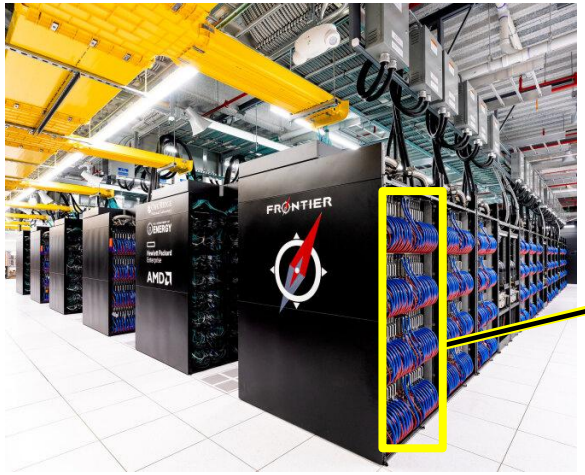
So the communication becomes the bottleneck, and AI accelerator utilization is degraded.

* TP: Tensor Parallelism, PP: Pipeline Parallelism, EP: Expert Parallelism

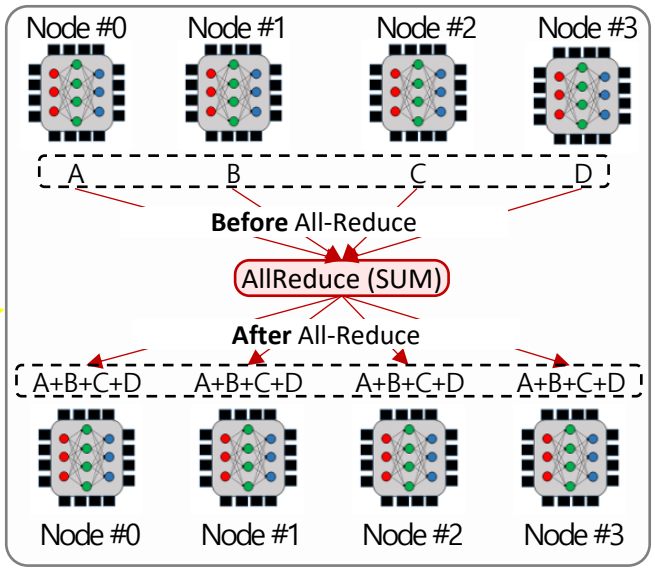
II. Background – Collective Communication Basics

So we need high-performance collective communication software to relieve these bottlenecks.

Collective communication is a communication pattern where a group of accelerators exchanges, combines, and shares data.



A large-scale cloud system equipped with AI accelerators



An AllReduce Example

Representative collective communication

- AllReduce
- AllGather
- ReduceScatter
- Scatter
- Gather

※ Each node has a partial sum, and after AllReduce every node has the same total sum.

II. Background – Collective Communication SW Trends

There are two types of CCL SW:
one is MPI-based CCL, and the other is vendor-specific CCL.

MPI-based CCL

MPICH, MVAPICH, Open MPI, Intel MPI, ...

- Supports heterogeneous systems and large-scale clusters
- Covers full distributed communication, including P2P

Vendor-specific CCL

NCCL, RCCL, OneCCL, HCCL, ...

- Optimized for a single vendor's accelerators
- Provides low-latency, high-bandwidth by utilizing the vendor's HW architecture

Trends in vendor-specific CCL

Global trends

- This vendor SW stacks can lead to the vendor lock-in problem.
- There is no general-purpose CCL that works across heterogeneous AI accelerators.

South Korea trends

- Current R&D projects focus mainly on per-chip performance.
- Multi-NPU support works only within a node.
- However, the heterogeneity across NPU/PIM/DPU products is rapidly growing.

* CCL: Collective Communication Library

II. Background – Motivation for KCCL

Problem — **Currently, there is no collective communication SW that integrates and optimizes South Korean AI accelerators.**

Low performance
There is no high-performance CCL SW for South Korean AI chips.

Low scalability
Scalability of the current CCL is limited for large-scale cloud systems.

Low resource utilization
It does not support heterogeneous AI chips.

Vendor lock-in
The high performance CCL SW stack depends on a single vendor.

Solution — **So we propose KCCL, a collective communication optimization SW for South Korean AI accelerators.**

High performance
It aims to deliver world-class CCL performance on South Korean AI chips.

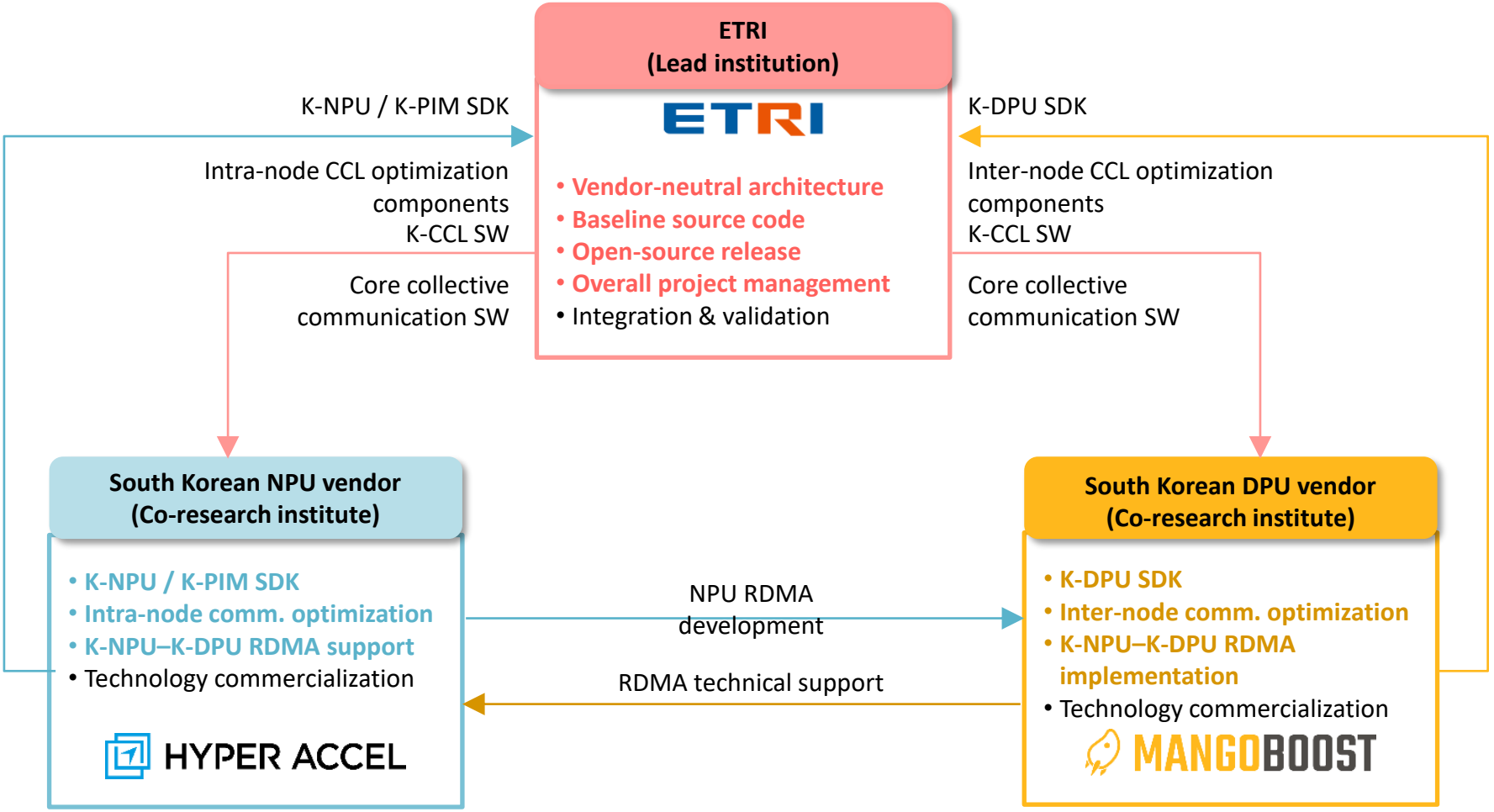
High scalability
It aims to scale stably as accelerators and nodes grow.

Maximized utilization
It aims to run CCL efficiently across NPU, PIM, and DPU.

Vendor neutrality
It aims to support diverse South Korean AI chips.

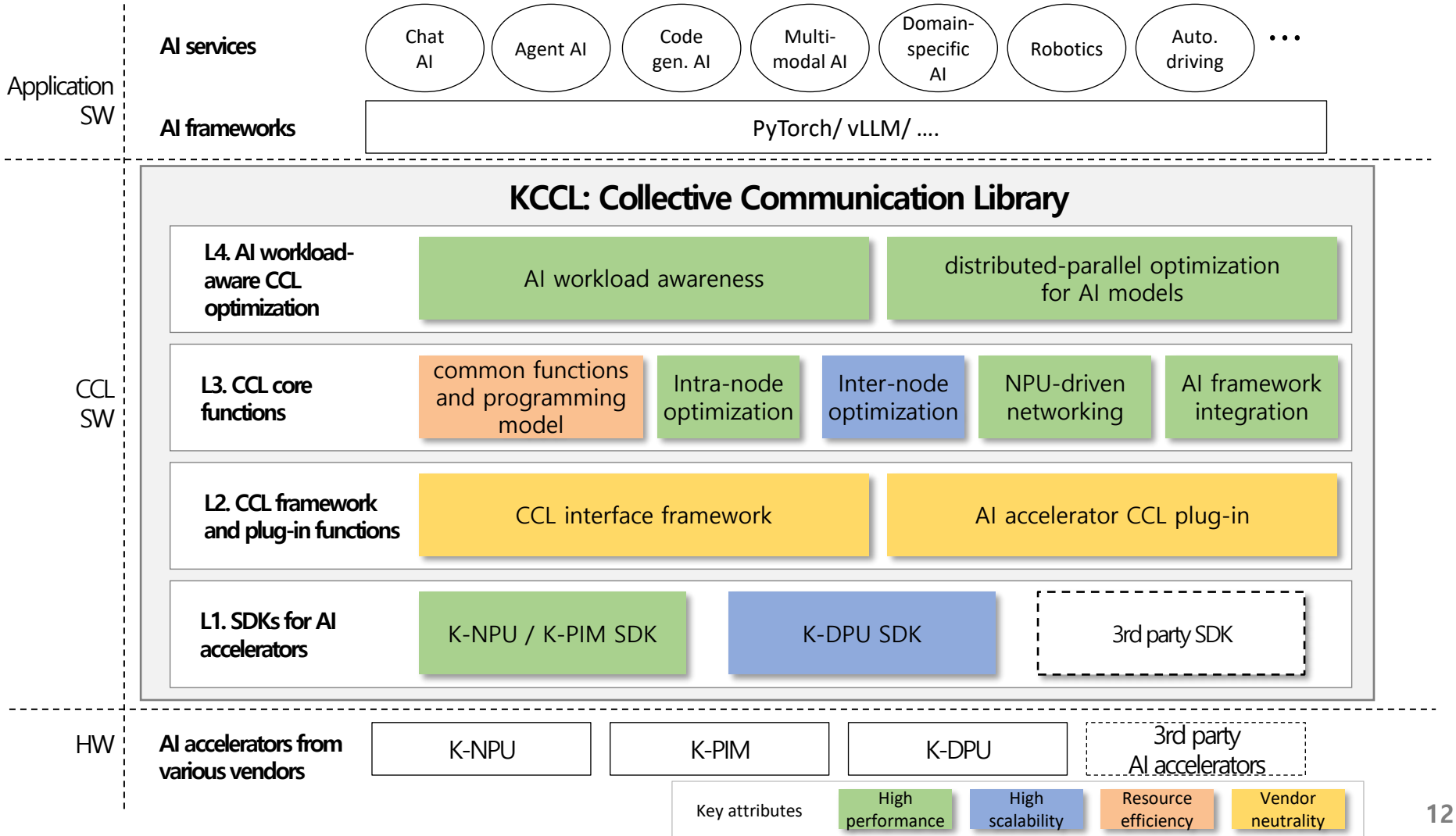
II. Background – Consortium and Execution Strategy

ETRI is the lead institution, and HyperAccel, a South Korean NPU vendor, and MangoBoost, a South Korean DPU vendor, are the co-research institute.



III. HW/ SW Architecture

We design a plug-in-based, vendor-neutral CCL architecture to support AI accelerators from multiple vendors.



IV. Research Outputs

K-NPU/ K-PIM SDK
providing low-level APIs
for intra-node collective communication

K-DPU SDK
providing low-level APIs
for inter-node collective communication

KCCL
a collective communication optimization SW for heterogeneous AI accelerators,
which is built by optimally combining these low-level APIs.

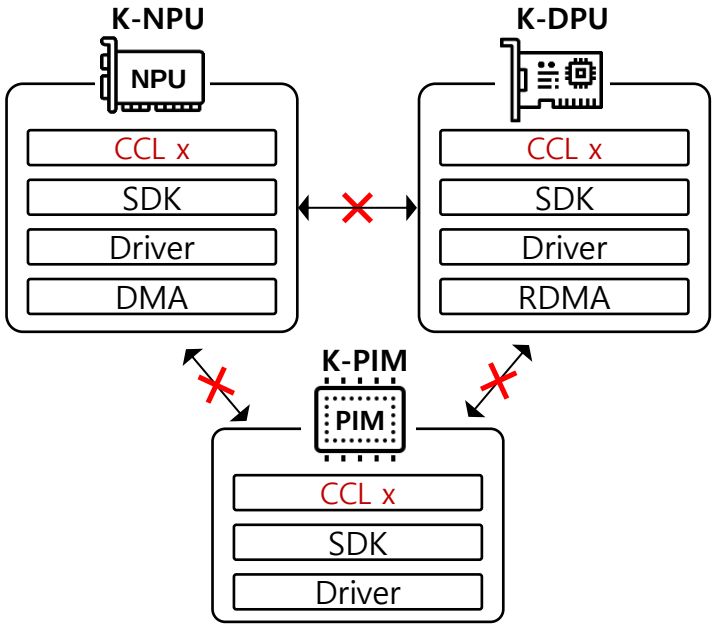
→ To achieve the goal of this research, we derived **6 research topics**.
Each topic will be explained from the next page.

V. Research Topic 1 | Resource Utilization Maximization Technology

Collective communication technology for joint optimization of data movement, computation, and synchronization across K-AI accelerators

(Conventional) Accelerator-specific communication

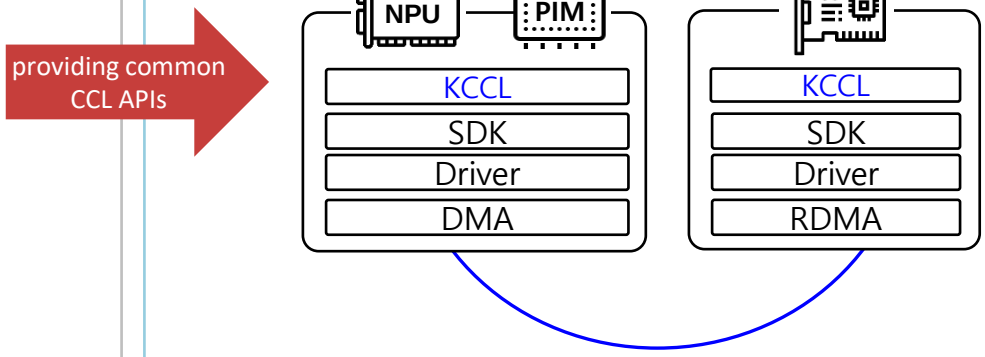
- The collective communication across heterogeneous K-AI accelerators is not possible



(Problem) Resource efficiency & system performance ↓

(Proposed) Accelerator-unified collective communication

- The unified CCL execution across heterogeneous AI accelerators will be possible, with CCL-level optimization
- HyperAccel has started developing a PIM-integrated NPU device in Korea



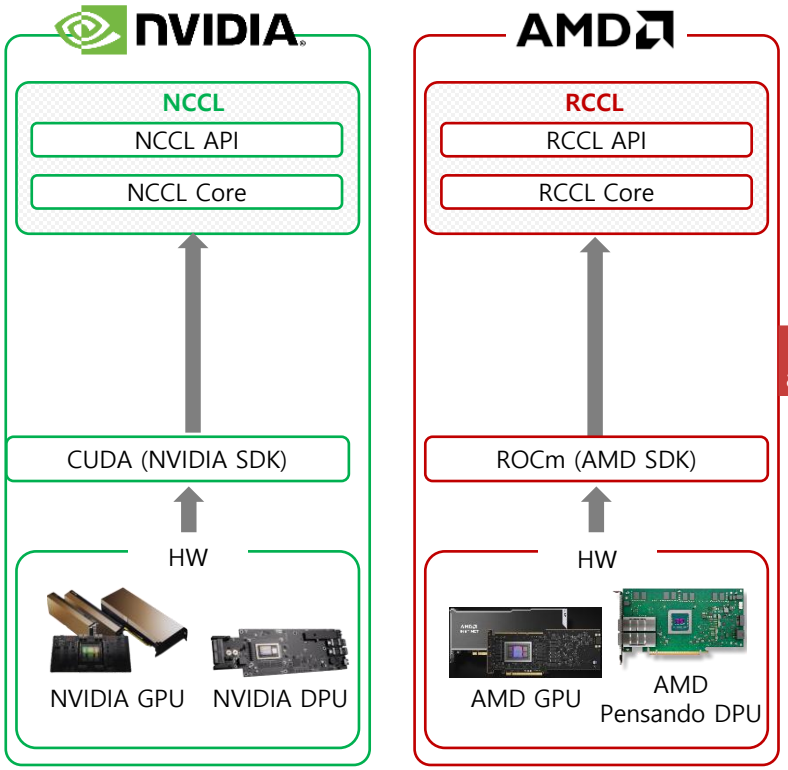
(Effect) Resource efficiency & system performance ↑

V. Research Topic 2 | Vendor-Neutral Collective Communication Technology

Collective communication technology based on a CCL framework and plug-in architecture

(Conventional) Vendor-dependent collective communication

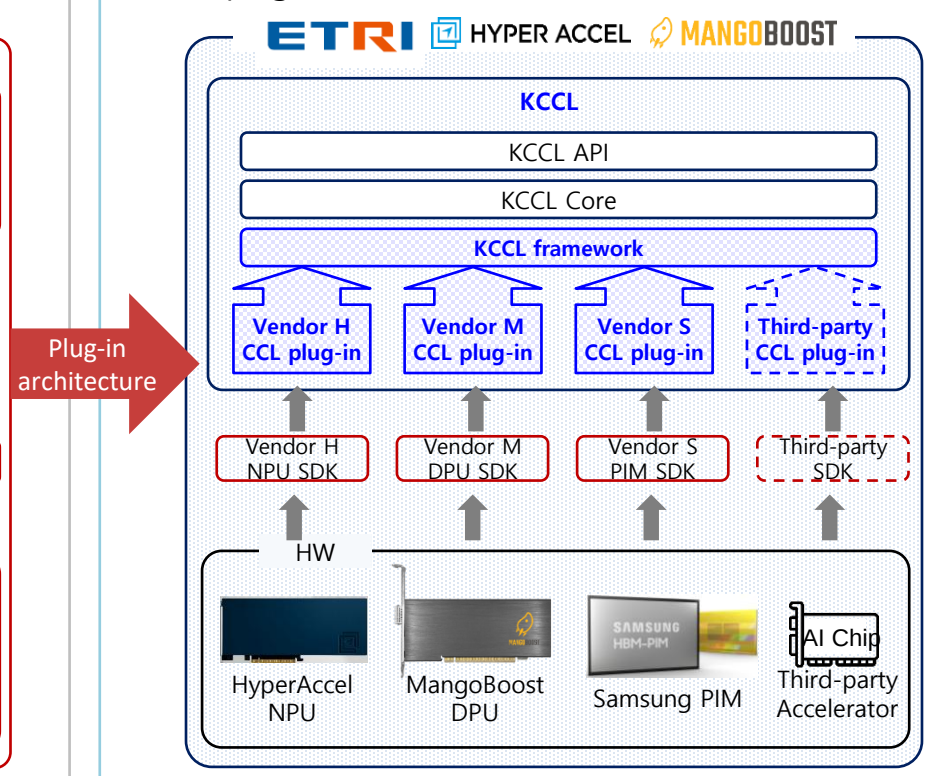
- Vendor-dependent runtime and CCL SW stack
- CCL SW re-implementation required for new accelerator



(Problem) Closed vendor ecosystem limits the AI infrastructure scalability ↓

(Proposed) Vendor-neutral collective communication

- Vendor-neutral runtime and CCL SW stack
- No CCL SW re-implementation required for new accelerators
- HW abstraction will be provided by the CCL framework and plug-in architecture



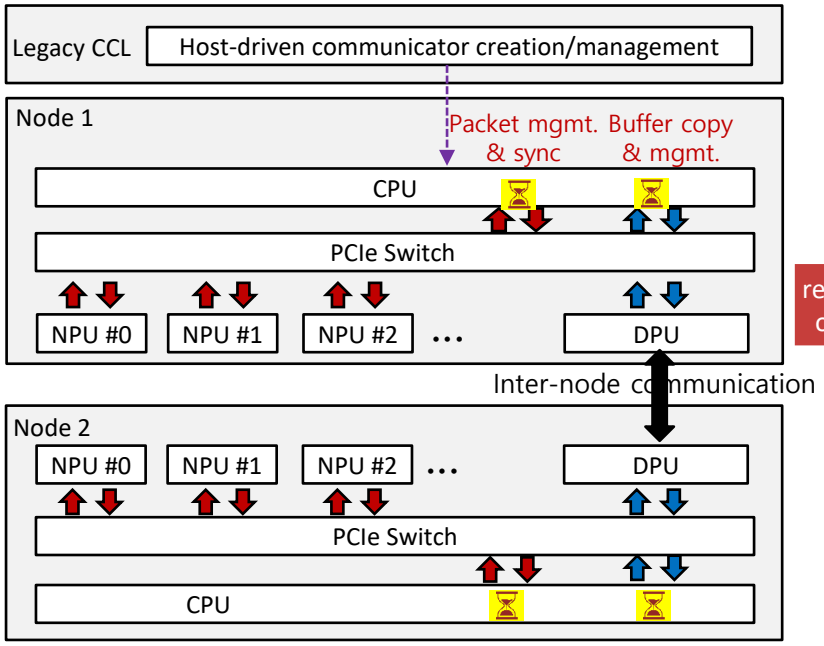
(Effect) Open vendor ecosystem improves the AI infrastructure scalability ↑

V. Research Topic 3 | Device-Driven Collective Communication Technology

Collective communication optimization technology that removes CPU-centric bottlenecks for intra- and inter-node communication

(Conventional) CPU-driven communication

- Intra-node communication
  - **CPU generates collective commands and manages device state**
  - Data detours device → CPU → device, adding latency
- Inter-node communication
  - **There are buffer copies through CPU memory**

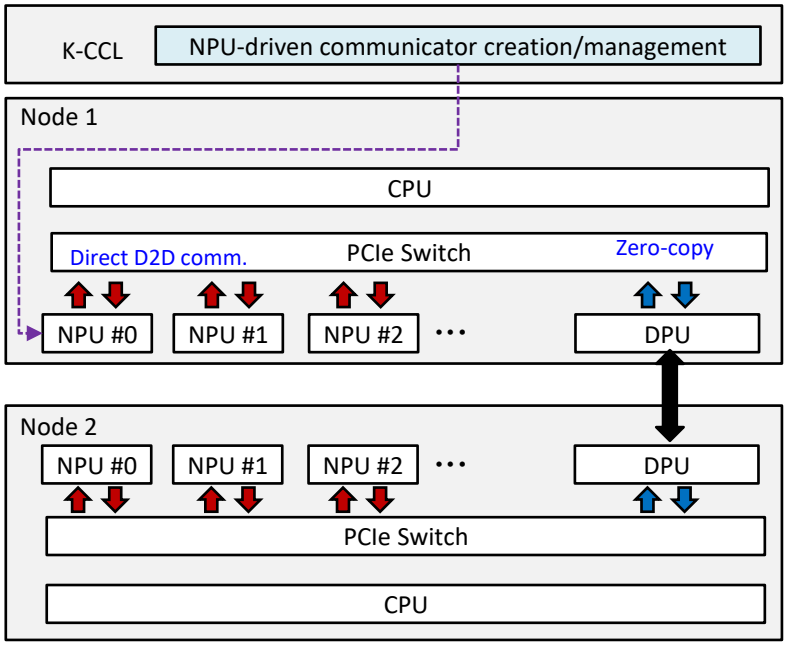


(Problem) Host overhead & communication latency ↑

reduce host overhead

(Proposed) Device-driven communication

- Intra-node communication
  - **Devices generate collective commands and manage device state**
  - Data moves directly device-to-device, lower latency
- Inter-node communication
  - **There are no buffer copies through CPU memory**



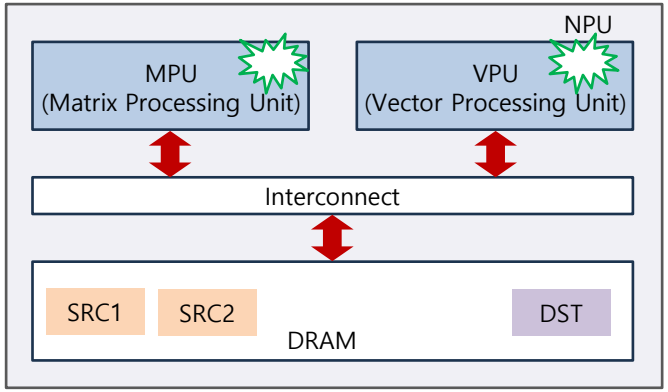
(Effect) Host overhead & communication latency ↓

V. Research Topic 4 | PIM-Based CCL Acceleration Technology

Collective operation offloading technology using PIM to minimize data movement

(Conventional) NPU-based collective communication

- Collective operations cause **frequent data movement between DRAM and NPU processors**
- **Large data transfers** are required for **simple reductions**
  - High-performance NPU units should be utilized for simple reductions

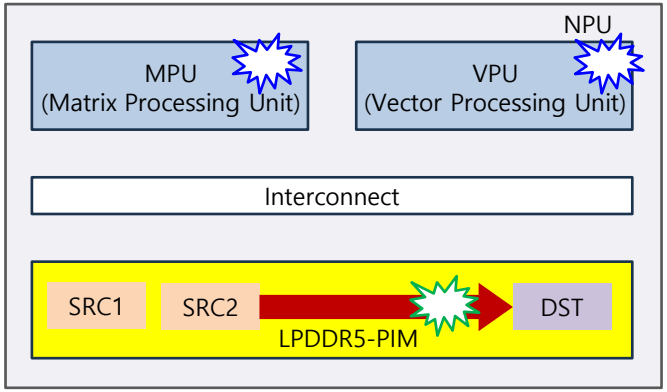


(Problem) Latency & power consumption ↑, system efficiency ↓

(Proposed) NPU–PIM-based collective communication

- Collective operations **no** longer cause **data movement between DRAM and NPU processors**
  - since reductions run inside the DRAM–PIM
- **Small data transfers** are required for **simple reductions**
  - High-performance NPU units can be utilized for AI inference jobs

PIM-accelerated reduction



(Effect) Latency & power consumption ↓, system efficiency ↑

Collective operation

Inference computation

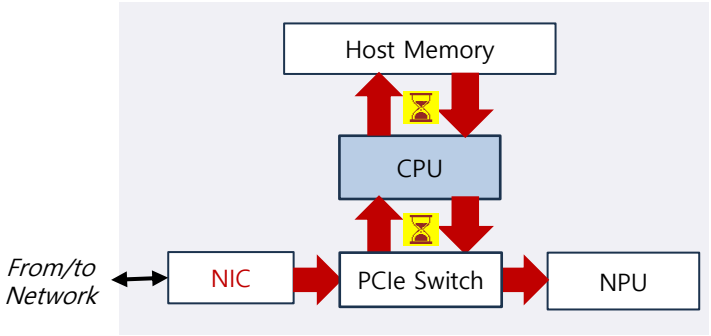
* PIM (Processing-In-Memory): performing computation inside memory instead of moving data to the compute unit

V. Research Topic 5 | P2P RDMA Technology Between DPU and NPU

Direct inter-node communication technology using K-DPUs

(Conventional) Host-memory-based RDMA communication

- **Indirect data transfer** is performed between **NIC** and **NPU**
  - NIC → host memory → PCIe switch → NPU
- **PCIe bandwidth is wasted**

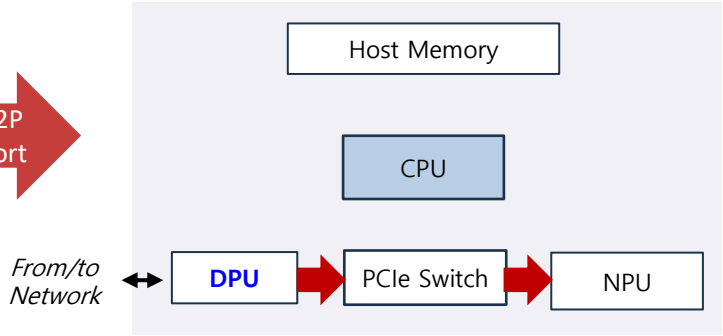


(Problem) Host overhead & latency ↑, bandwidth efficiency ↓

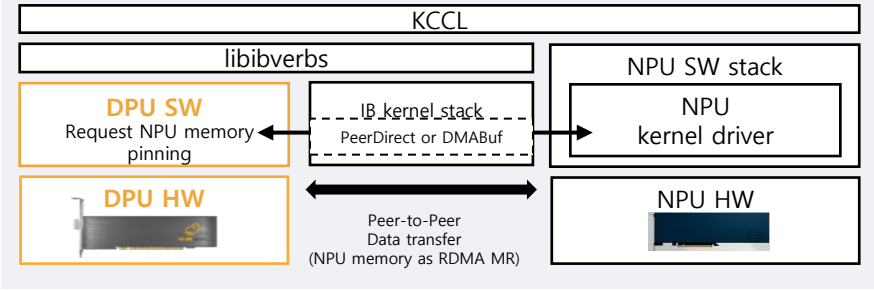
NPU/DPU P2P RDMA support

(Proposed) Direct DPU–NPU P2P RDMA communication

- **Direct data transfer** is performed between **NIC** and **NPU**
  - Peer-to-peer DMA between DPU and NPU
- **PCIe bandwidth is not wasted**



(Effect) Host overhead & latency ↓, bandwidth efficiency ↑



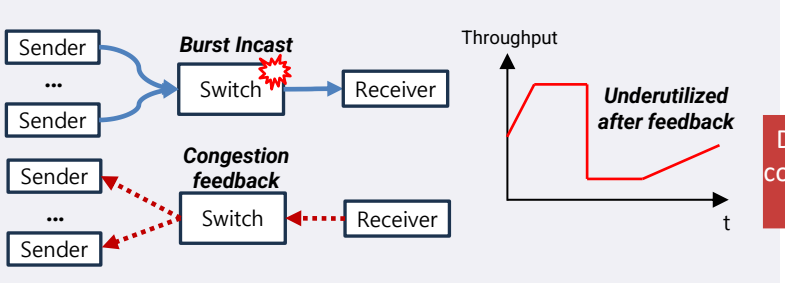
※ This technology will be designed to be compatible with the standard RDMA software stack.

V. Research Topic 6 | Network Congestion Control Technology

Inter-node communication optimization technology based on UEC standards

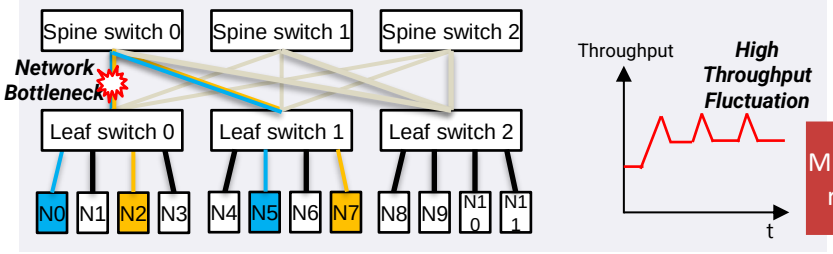
(Conventional) Static network control approach

- **Fixed congestion control**
  - Limited adaptation to changing network conditions (e.g., DCQCN)



* DCQCN: a RoCE-based data center congestion control algorithm

- **Fixed-path routing**
  - Traffic congestion concentrates on specific paths
  - e.g., ECMP-based routing



* ECMP (Equal-Cost Multi-Path): multi-path traffic distribution routing

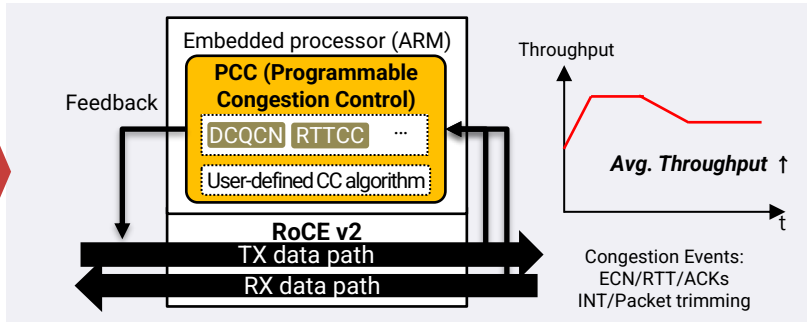
(Problem) Network utilization ↓, latency ↑

Dynamic congestion control

Multi-path routing

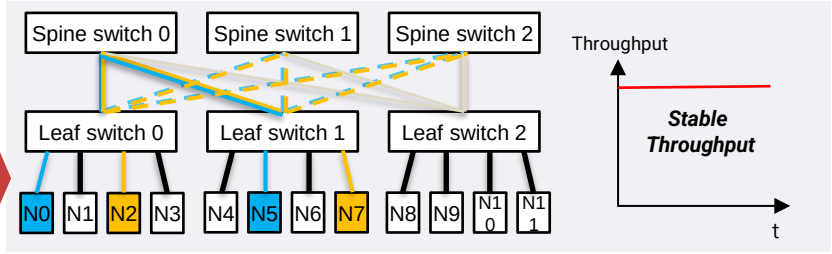
(Proposed) Dynamic network control approach

- **Dynamic congestion control**
  - User-defined dynamic congestion control (e.g., PCC)



* RTTCC (Round Trip Time Congestion Control): RTT-based dynamic network congestion control

- **Multi-path routing with packet spraying**
  - Congestion mitigation through per-packet path distribution
  - e.g., packet-spraying-based routing



(Effect) Network utilization ↑, latency ↓

* UEC (Ultra Ethernet Consortium): next-generation high-performance Ethernet standards for AI/HPC data centers

Network congestion

VI. Conclusion

- The Korean government has launched the development of KCCL, a collective communication SW that supports not only K-AI accelerators but also third-party accelerators

- We have just begun this five-year journey for KCCL, and we look forward to your interest and support

Contents

I. Research Overview

Item	Description
Project Title	Enhancing MPI Collective Communication Performance Using CXL Shared Memory and an Intelligent CXL Switch
Organization	ETRI, OSU (The Ohio State University) Joint Research
Period	2 year (From September 2023 through Aug. 2025)
Goal	To improve the MPI inter-node collective communication performance in a multi-node environment connected by CXL

ETRI

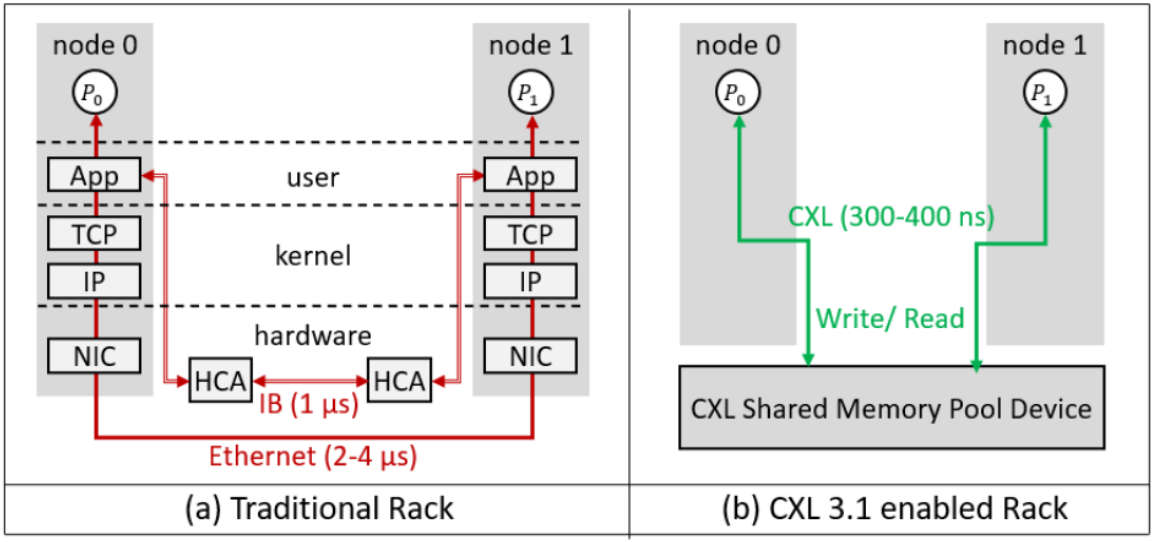
- **Focus: Single Rack-Scale CXL Memory Pool / Intelligent CXL Switch**
- Approach 1. CXL SHM-Based AllGather
- Approach 2. iMEX-Based Collective Communication (Intelligent CXL Switch-Based)

The Ohio State University

- **Focus: Beyond Rack-Scale CXL Memory Pool**
- Approach 1. Improving collective communication performance using a beyond rack-scale CXL memory pool device
- Approach 2. Demonstration applications showcasing CXL-based collective communication

II. Motivation

- In a traditional rack using Ethernet and InfiniBand, the communication latency between processes running on remote nodes is 2–4 µs and about 1 µs, respectively.
- However, in a CXL 3.1-enabled rack, that latency can be reduced to 300–400 ns.



Comparison of Inter-Node Communication Latency

III. Proposed Approach

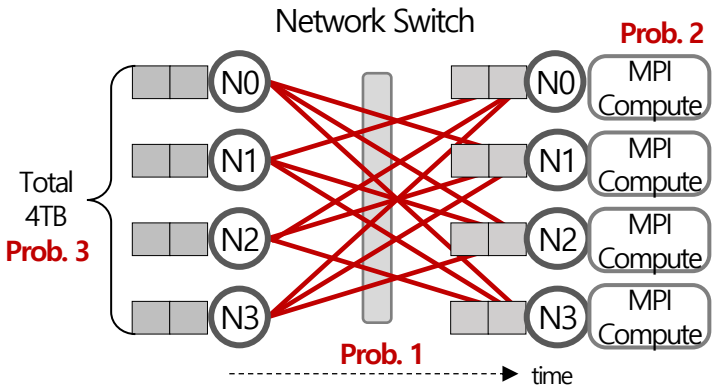
- We proposed iMEX (intelligent CXL-based Memory EXpander)

Data-Intensive Applications of **AI** and **HPC** fields

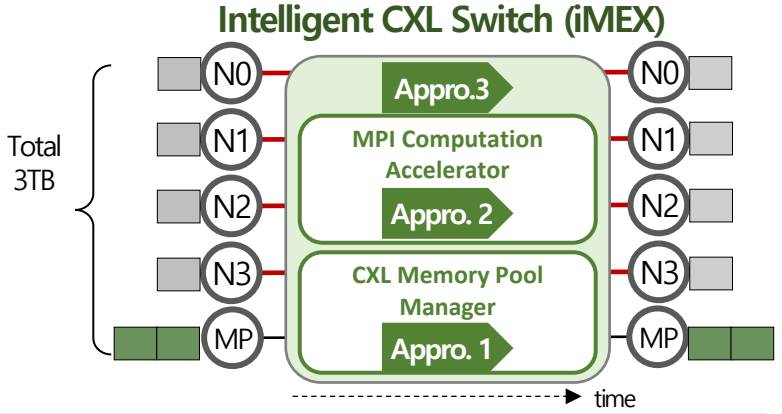
※ **Appro.3.** MVAPICH2 optimized for iMEX

All-Reduce

(AS-IS) Conventional Collective Communication



(TO-BE) Proposed Collective Communication



• # of Communication : n^2 (pairwise comm.) • Communication **Latency** : 2-4 μs (100 GE)	Communication	• # of Communication : n • Communication **Latency** : 300-400 ns (CXL Switched DDR)
• MPI Computation performed on **CPU** • # of Computation : n	Computation	• MPI Computation performed on **Dedicated Accelerator** • # of Computation : **1**
• **Low**	Main Memory Utilization	• **High**

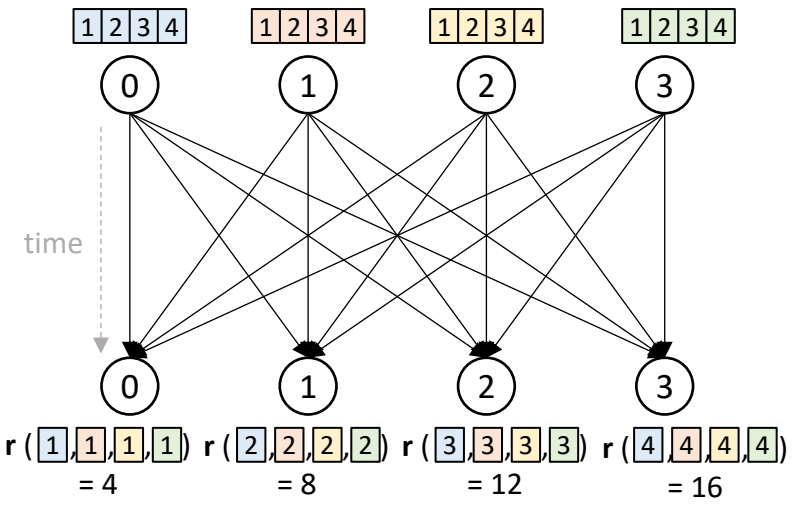
Main memory (512GB) N Computing Node : Data Movement

CXL memory pool (512GB) MP Memory Pool n : # of Processes

IV. Intelligent CXL Switch-based Collective Communication – Design

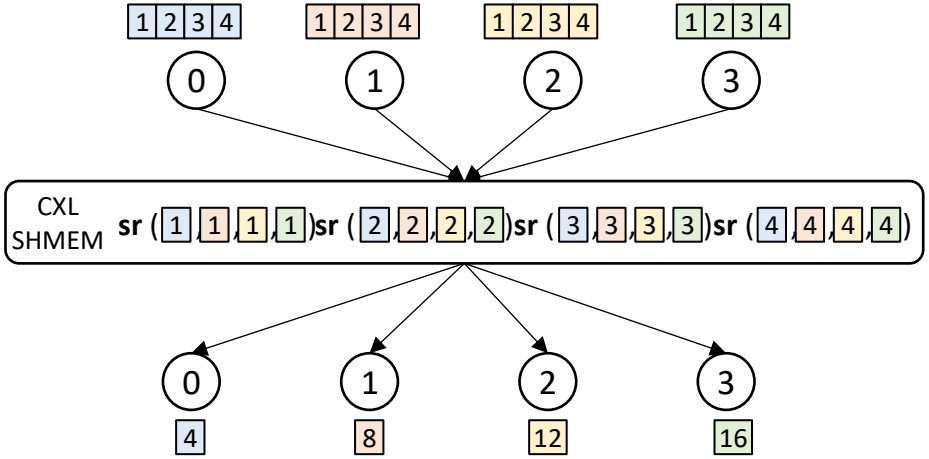
- We design and implement the
  - iMEX CXL Switch, which consists of an MPI operation accelerator
  - reducescatter and AllReduce utilizing iMEX

Conventional. Network-Based Collective Communication



(e.g., ReduceScatter, sum)

Proposed. iMEX-Based Collective Communication



(e.g., ReduceScatter, sum)

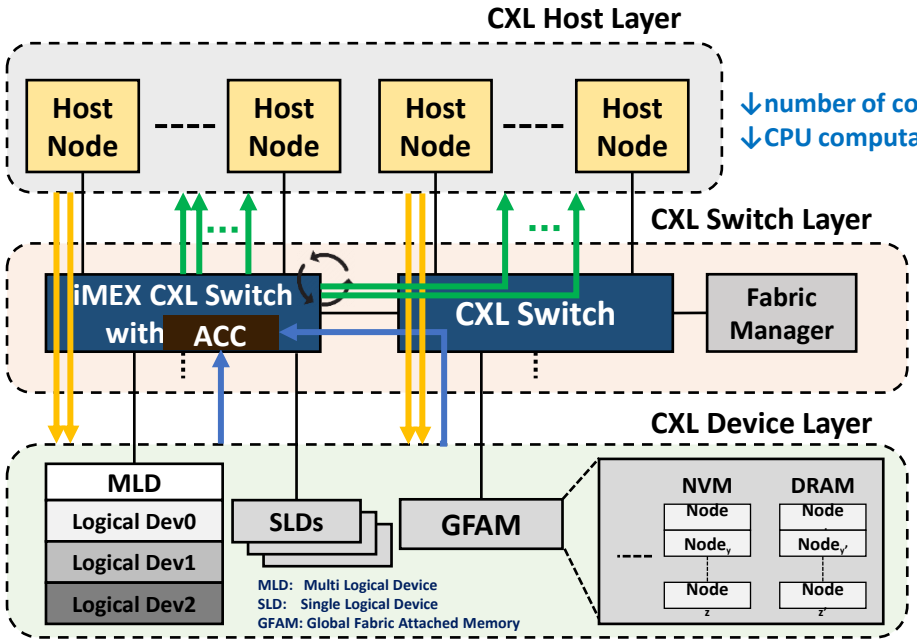
※ **sr**: intelligent CXL switch's reduction operation

Expect performance improvement by reducing
- the number of communications
- the CPU overhead by offloading MPI computation to the switch's accelerator

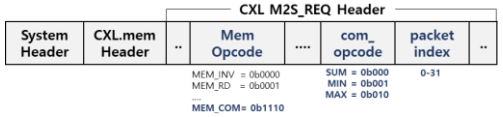
※ r : CPU reduction operation ※ sr : CXL switch reduction operation

IV. Intelligent CXL Switch-based Collective Communication – Implementation

- We define the offloading and result packet structures of the iMEX CXL switch
- We extend MVAPICH2 2.3.7 to leverage the iMEX CXL switch
- This research was accepted at IEEE CLUSTER 2026 * and will be presented next month



iMEX-Based Collective Communication Architecture



Offloading Packet Structure

Result Packet Structure

Issuing offloading command

Gathering data into the accelerator

Collective operation on the accelerator (e.g., SUM/MIN/MAX)

Broadcasting collective operation result

ACC MPI communication and computation accelerator

*SEON YOUNG KIM, et al. "CoCoA: An In-CXL Switch Collective Communication Accelerator." 2026 IEEE CLUSTER. (Accepted)

V. Conclusion

- We improved memory utilization for AI and HPC systems by using the CXL memory pool
- We enhanced collective communication performance with the MPI accelerator in the iMEX CXL switch
- Through these two approaches, we developed technologies that can contribute to improving the performance of AI and HPC applications

Thank you

Contact: ahnhy@etri.re.kr