k-NN Accelerator using Near Memory Processing and MPI Technology

ETRI

Supercomputing Technology Research Center

August 23, 2023

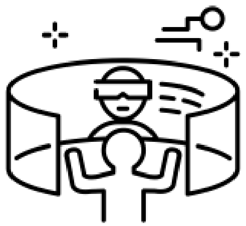
HooYoung Ahn

Contents

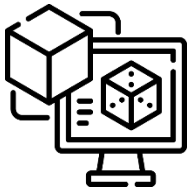
- Motivation
- Problem Definition
- Our Approach
- Demonstration
- Roadmap

Motivation

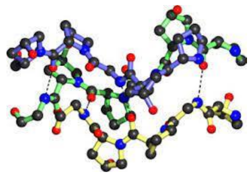
- Worsening memory wall problem
  - The **memory wall** problem refers to the increasing speed gap between the CPU and memory, commonly known as the memory latency
  - This problem is becoming more serious in **data-intensive applications in AI and HPC**
    - ✓ These applications incur **frequent memory access** and **heavy I/O** to process large-scale data

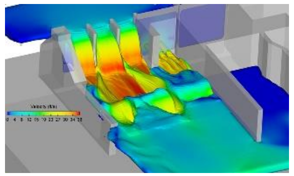

Metaverse


Autonomous Driving


Digital Twin


Graph Analysis

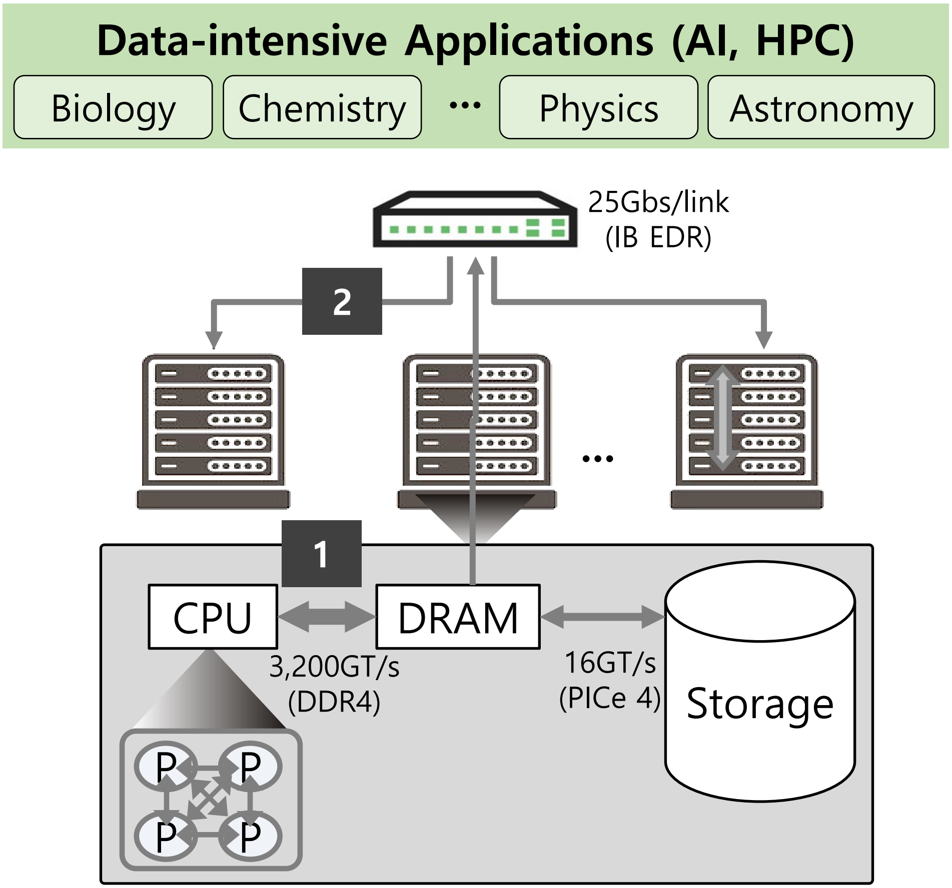

Protein-Structure Analysis


Computational Fluid Dynamic Simulation

- We aim to address this problem by utilizing **a large memory expander with acceleration**

Problem Definition

- The memory wall gets worse in Data-intensive Applications
  - Because processing those applications in parallel across multiple servers brings two problems



1 **In a single server,**

As CPU performance increases,

- the ***limitations*** in ***memory capacity*** and ***memory bandwidth*** **degrades the system performance**

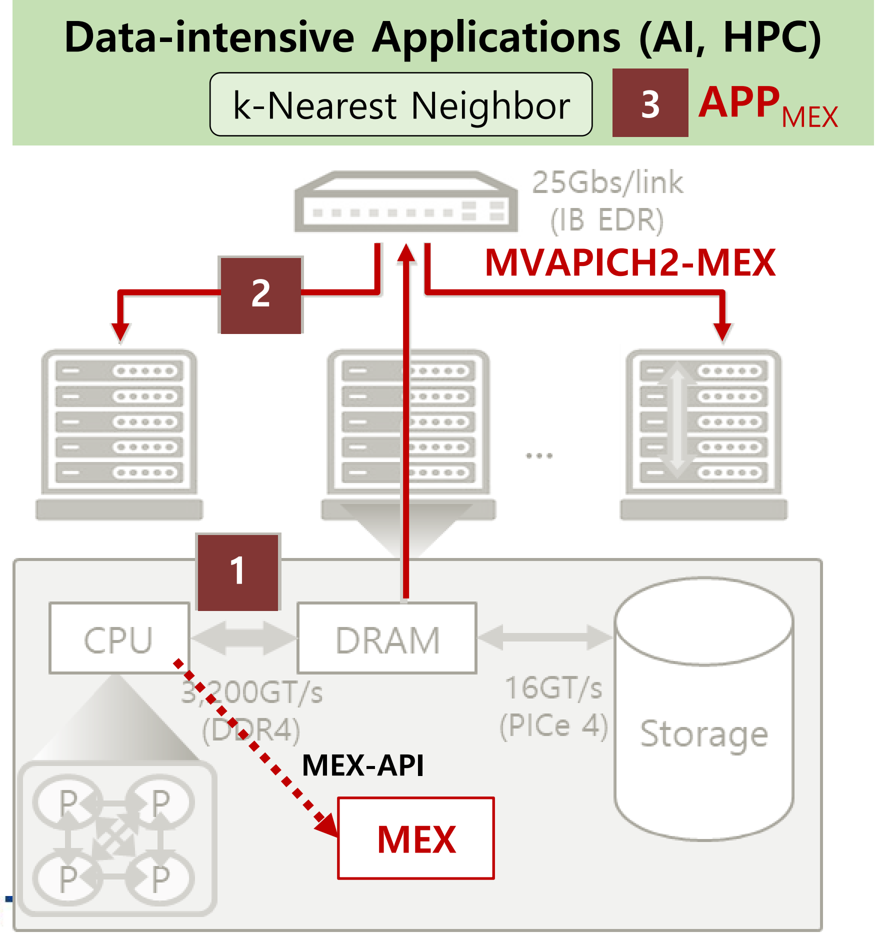
2 **Between multiple servers,**

As the number of nodes increases,

- ***a data movement bottleneck*** that arises between the nodes **degrades the system scalability**

Our Approach

- To solve the problems, we propose MEX, MVAPICH2-MEX, and APP_{MEX}
  - MEX is the device with a large memory and MPI collective communication accelerator
  - MVAPICH2-MEX is the optimized version of MPI library for MEX
  - APP_{MEX} is the data-intensive application using MEX



1 Utilize the **large memory of MEX**

2 Utilize the **MVAPICH2-MEX**
→ **Near-Memory Processing** of MPI collective communication
→ Reduce communication overhead

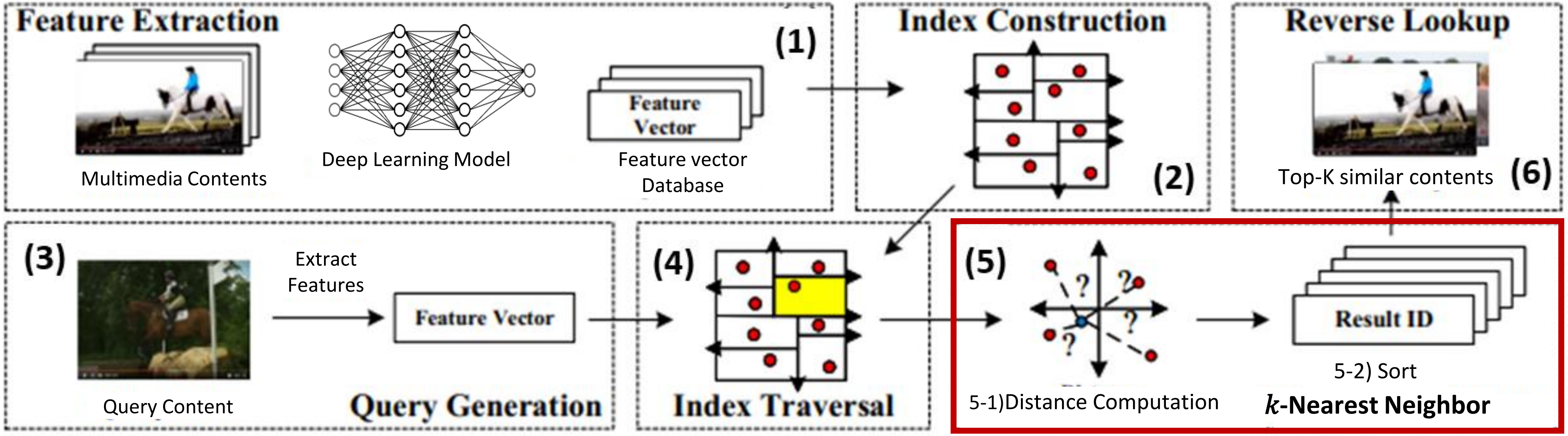
3 Verify our approach using target use case application APP_{MEX}

APP_{MEX}| Similarity Search

- Data-intensive applications across various AI and HPC fields
  - Natural Language Processing [3]
  - Genome Analysis [4]
  - Graph Similarity Search [5]
  - Molecular Similarity Search [6]
  - Image Similarity Search [2]

  } use k-NN

- Target use case

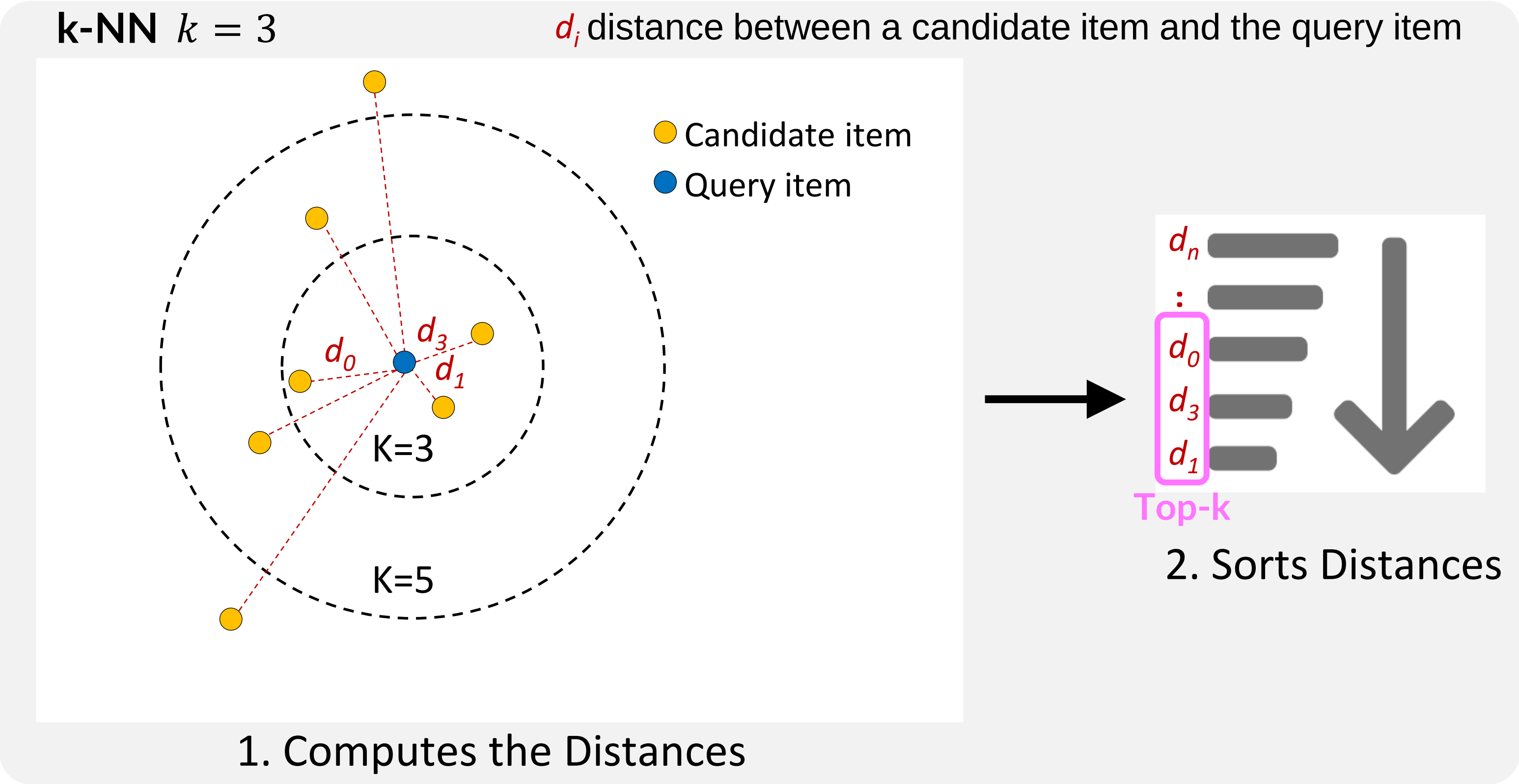


Offline : (1), (2) Online : (3)~(6)

Overall flow of the similarity search in multimedia database [7]

APP_{MEX} | k-NN in Similarity Search

- ***k*-NN** is an operator to find the ***k*** items closest to a given query [8]



The execution process of the k-NN operator

$\mathrm{APP}_{\mathrm{MEX}}$ | k-NN in Similarity Search

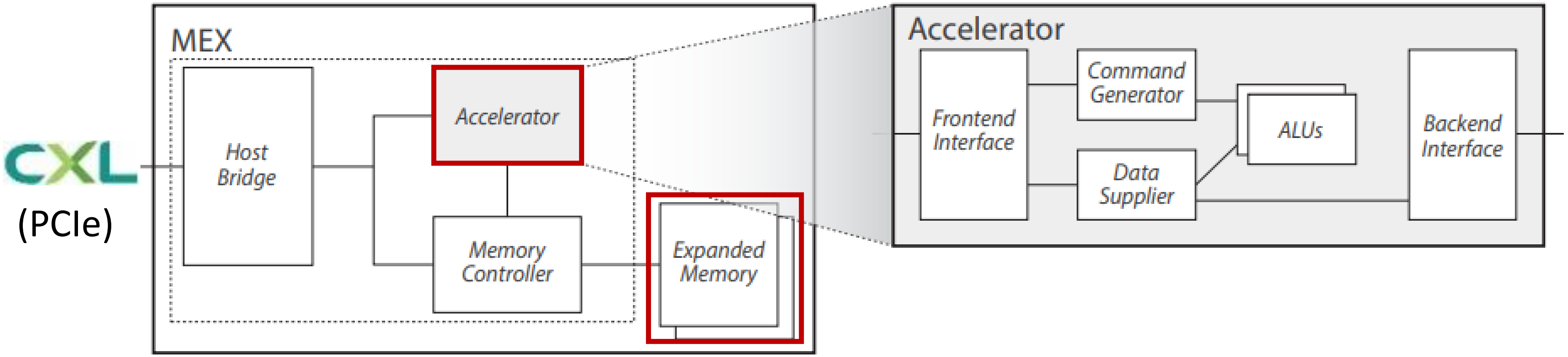
- k-NN is suitable for offloading to MEX
  1. **Distance computation** between multi-dimensional vectors can greatly benefit from massive parallelism (Compute-intensive)
     → benefit from MEX accelerator
  2. Additional memory is needed to store intermediate **Sort** results (Memory-intensive)
     → benefit from MEX memory
  3. Parallel sort outperforms the serial **Sort** for large datasets (Compute-intensive)
     → benefit from MEX accelerator

- Therefore, we have decided to offload the k-NN operator to MEX

※ Since MEX is ***a device*** with ***large memory*** and ***computational capabilities***, ***more compute-intensive and memory-intensive tasks*** have substantial potential for performance improvement when utilizing MEX.

Memory EXpander (MEX)

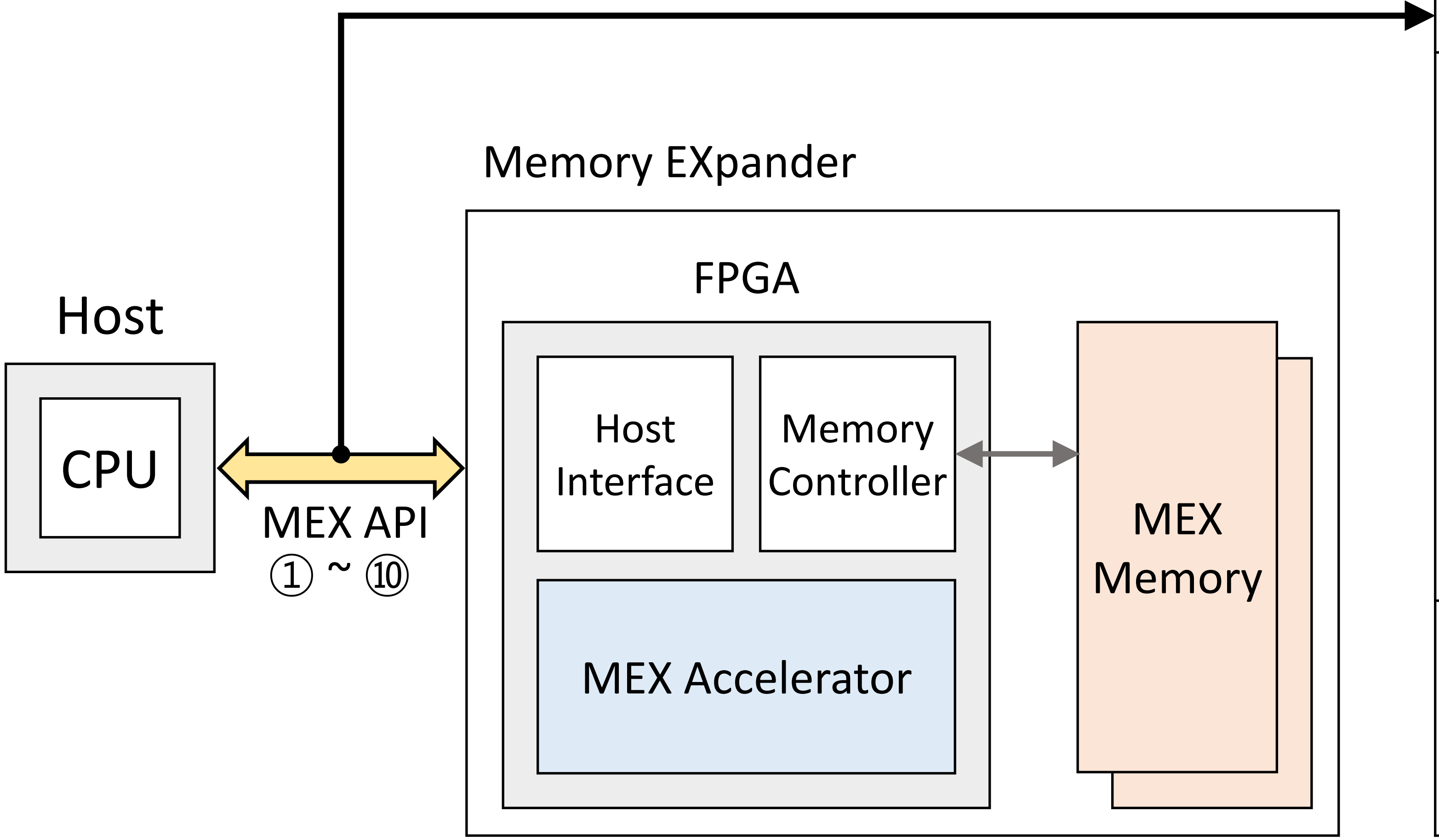
- MEX is an on-board device that provides
  - **Large memory** capacity
  - MPI collective communication **accelerator**
    - ✓Performs the ***Near-Memory Processing*** of MPI collective communication [1]
  - Connected to the host server through **CXL (Compute Express Link)**



The simplified MEX Architecture [9]

MEX & MEX API

- Hosts can use MEX as a ***memory semantic*** way, utilizing the ***MEX API***

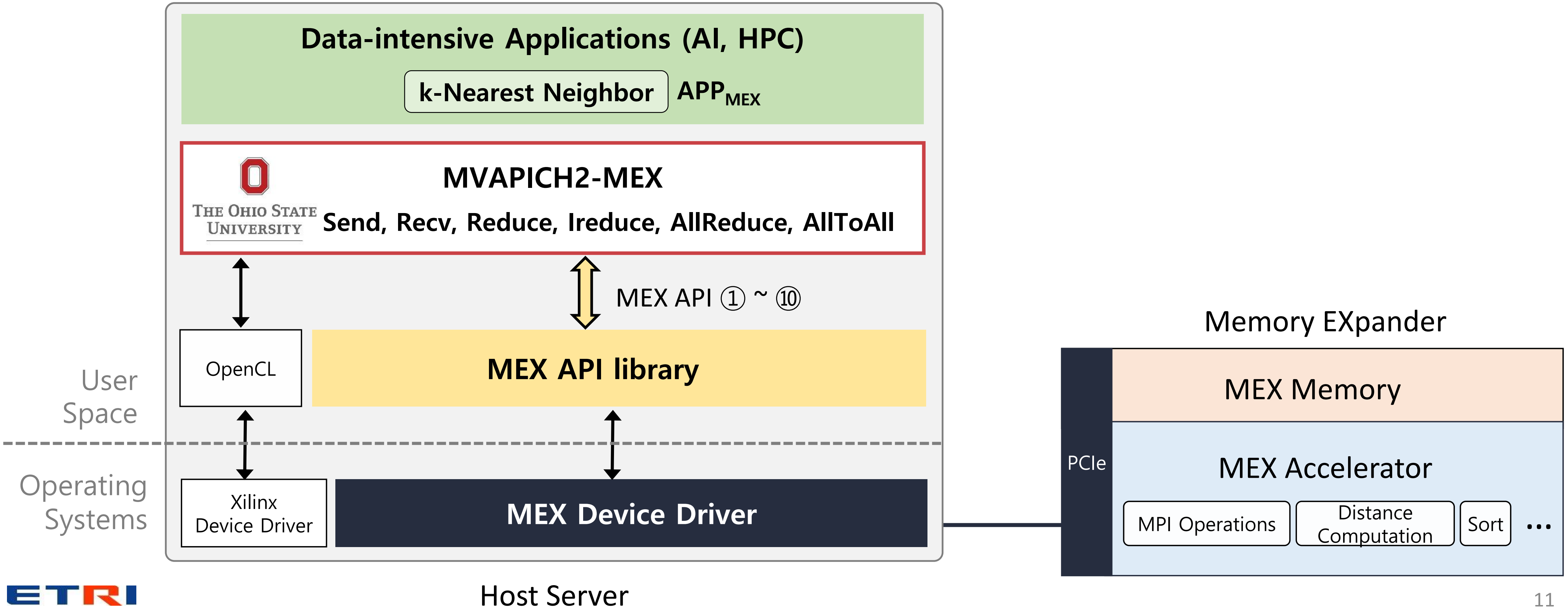


MEX API	
MEX Mem. API	① MEX_Init ()
	② MEX_Alloc ()
	③ MEX_Write ()
	④ MEX_Read ()
	⑤ MEX_Memset ()
	⑥ MEX_Free ()
	⑦ MEX_finalize ()
MEX Acc. API	⑧ MEX_DistanceAndSort ()
	⑨ MEX_Reduce ()
	⑩ MEX_Wait ()

System Architecture of k-NN accelerator

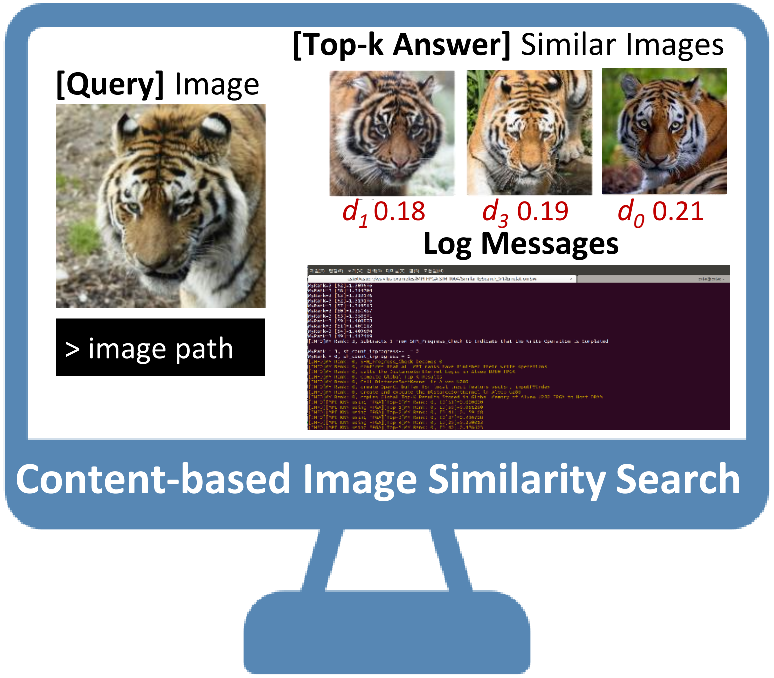
- MVAPICH2-MEX
  - uses **MEX memory** as a **communication buffer**
  - uses **MEX accelerator** as a **collective communication accelerator**

  → enable the ***Near Memory Processing*** of the MPI collective communications



Demonstration

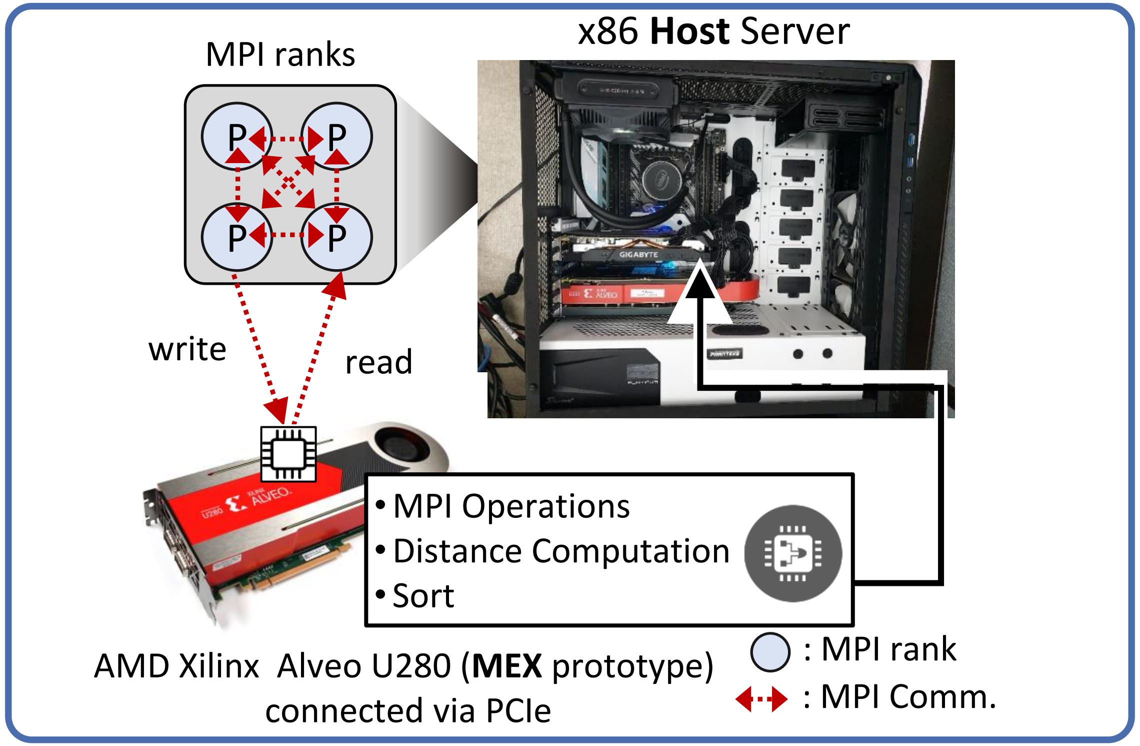
MEX memory MEX accelerator

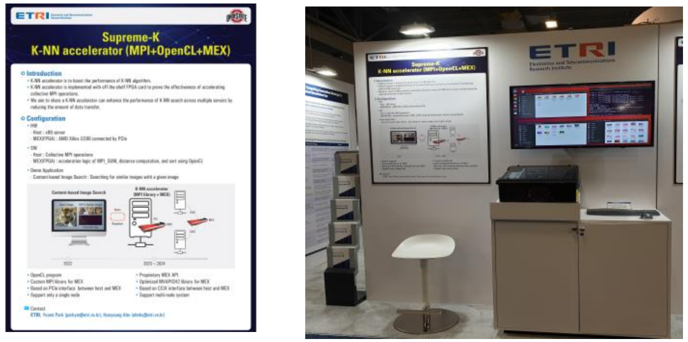


Query

Response

k-NN Accelerator using MPI and MEX





Brochure and Booth of SC22

	SC22	SC23
Host SW	• OpenCL • MPI : Custom MPI library, Blocking MPI communication	• Proprietary **MEX API** • MPI : **MVAPICH2-MEX,** both **Blocking** and **Non-blocking** MPI comm.
MEX HW	C++ HLS-based kernel accelerator	**RTL**-based kernel accelerator
Dataset	Color histogram feature vectors	**Deep Learning** feature vectors

System Configuration and Dataset

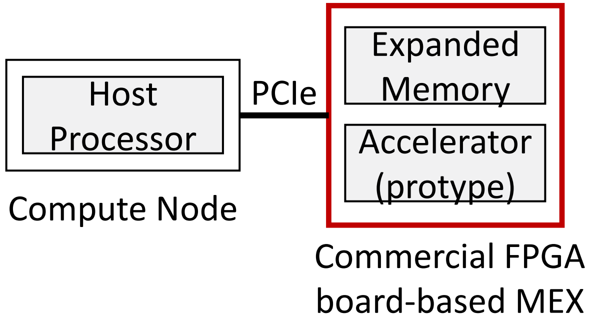
Demonstration

MEX Roadmap

- We aim to improve the performance of ***data-intensive applications*** in ***multi-node systems***
- Utilize CXL to improve ***both*** the ***scale-out*** and the ***scale-up*** performance

1

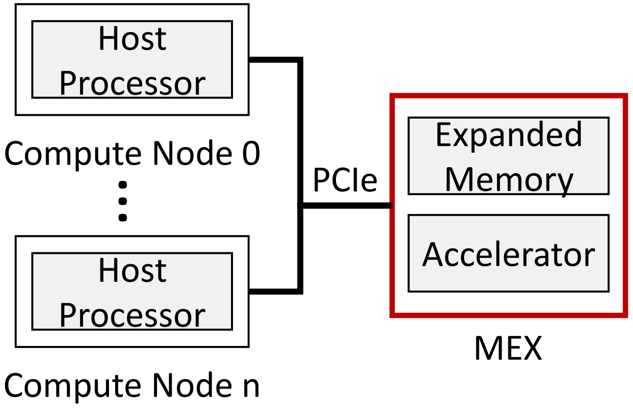
- Commercial FPGA board-based MEX
- Up to 32GB expanded memory
- Prototype version of accelerator
- Support only a single node



Now, we are here

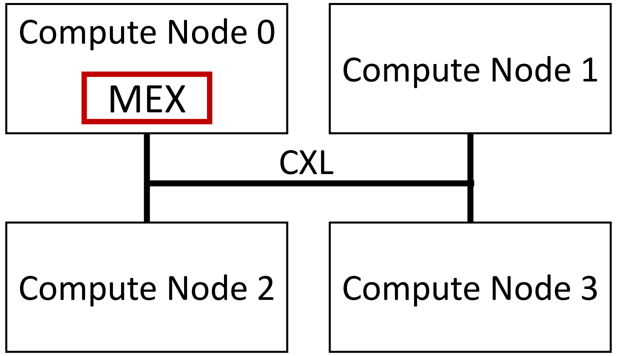
2

- MEX with our own board design
- Up to half TB of expanded memory
- Full-featured accelerator
- Support multi-node system



3

- Support multi-node using CXL
- Further performance improvement in MPI collective communication



4

- MEX SoC (System on Chip)

ETRI

Reference

1. Kim, S. Y., Ahn, H. Y., Jun, S. I., Park, Y. M., & Han, W. J. (2022). Trends of the CCIX Interconnect and Memory Expansion Technology. Electronics and Telecommunications Trends, 37(1), 42-52.

2. Istrate, Daniela, Alina Bora, and Luminita Crisan. "A first attempt to identify repurposable drugs for type 2 diabetes: 3D-similarity search and molecular docking." Chemistry Proceedings. Vol. 3. No. 1. Multidisciplinary Digital Publishing Institute, 2020.

3. Helmers, Lea, et al. "Automating the search for a patent’s prior art with a full text similarity search." PloS one 14.3 (2019): e0212103.

4. Ayyad, Sarah M., Ahmed I. Saleh, and Labib M. Labib. "Gene expression cancer classification using modified K-Nearest Neighbors technique." Biosystems 176 (2019): 41-51.

5. Chang, Lijun, et al. "Accelerating Graph Similarity Search via Efficient GED Computation." IEEE Transactions on Knowledge and Data Engineering (2022).

6. Antelo-Collado, Aurelio, et al. "Maximum common property: a new approach for molecular similarity." Journal of cheminformatics 12.1 (2020): 1-22.

7. Lee, Vincent T., et al. "Application codesign of near-data processing for similarity search." 2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2018.

8. Seidl, Thomas, and Hans-Peter Kriegel. "Optimal multi-step k-nearest neighbor search." Proceedings of the 1998 ACM SIGMOD international conference on Management of data. 1998.

9. Kim, A. S. Y., Ahn, A. H. Y., & Park, A. Y. MEX: CXL-Based Memory EXpander With Hardware Acceleration.

Thank You!

Contacts : ahnhy@etri.re.kr