K-NN accelerator (MPI+OpenCL+MEX)

2022.11.16

HooYoung Ahn

ahnhy@etri.re.kr
Background & Motivation

- Data-intensive applications
  - **Definition**
    - ✓ Data analysis and AI applications that increase the performance and accuracy by using large-scale data in HPC fields
  - **Examples**
    - ✓ Weather forecasting, protein structure analysis, autonomous driving, digital twin, etc.
  - **Requirements**
    - ✓ need a large memory and accelerators for the fast parallel processing of big data

---

![Computer Architecture Diagram](image)

- **As Is**
  - Limited number of memory slots
- **To Be**
  - Extended form of memory with accelerator outside the node
Memory Expander (MEX)

- An on-board device that provides
  - additional **memory capacity**
  - **acceleration capabilities** to enhance the performance of parallel processing workload
    ✓ by offloading specific MPI collective APIs such as MPI-Reduce and MPI-AllReduce to MEX [1]
  - Connected to the host server using CCIX/ CXL
MEX use case | Similarity Search

- **Definition**
  - The task of retrieving items that are similar to a given query

- **Applications**
  - Content-based image search of 3D models [2]
  - Natural Language Processing [3]
  - Genome Analysis [4]
  - Graph mining [5]
  - Molecular Similarity Search [6]

Overall flow of the similarity search in Multimedia Database [7]
MEX usecase | K-NN in Similarity Search

- **K-NN** is an operator to find the $k$ items closest to a given query [8]
- Makes a large workload by
  - computing the distance between all candidate items and the query item
  - sorting the set of distances

---

**K-NN $k = 3$**

$d_i$ distance between all candidate items and the query item

1. **Distance Computation**

2. **Sort**

Execution process of the K-NN operator
MEX usecase | K-NN in Similarity Search

- K-NN satisfies the **offloading suitability** well
  - Distance computation
    ✓ Computation between multi-dim. vectors can benefit from massive parallelism
      → It is the **Computation Intensive task**
  - Sort
    ✓ Needs additional memory to store the intermediate sort results
      → It is the **Memory Intensive task**
    ✓ Parallel sort is faster than serial sort for large datasets
      → It is the **Computation Intensive task**

- We decide to offload the K-NN operator to MEX

*Offloading Suitability for MEX*

- The large offloading suitability means that there is much room for performance improvement by using the additional memory capacity and accelerator provided by MEX
- That is, the more **computation intensive** and **memory intensive** tasks, the greater the offloading suitability
**K-NN accelerator** is a device that accelerates MPI-based K-NN using MEX.

### Application
Content-based Image Similarity Search

**[Query] Image**

**[Top-k Answer] Similar Images**

- \( d_1 = 0.18 \)
- \( d_2 = 0.19 \)
- \( d_0 = 0.21 \)

**Log Messages**

Exhibiting @KISTI booth (#839)

### HW
K-NN accelerator using MEX (FPGA) x86 Host Server

- MPI ranks
- write
- read

- AMD Xilinx Alveo U280 (MEX prototype)
- Connected by PCIe
-MPI_OPs
- Distance Computation
- Sort

### SW

<table>
<thead>
<tr>
<th>Host</th>
<th>Custom MPI library for MEX, OpenCL program</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dataset: 128 RGB images in 10 categories</td>
</tr>
<tr>
<td></td>
<td>with large color-diff. (cherry blossom, tiger, …)</td>
</tr>
</tbody>
</table>

- MEX
  - Acceleration logic (OpenCL kernel) of MPI-SUM, MPI-MAX, Distance Computation, Sort

HW and SW system configuration
K-NN accelerator | Development Environment

- **Host Server**
  - CPU: Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz
  - Main Memory: 64 GB
  - OS: Ubuntu 18.04.6 LTS
  - Libraries: MVAPICH2 version 2.3.7, OpenCV 4.6.0

- **FPGA (MEX prototype)**
  - AMD Xilinx® Alveo™ U280 Data Center accelerator cards
    - Memory: 32GB DRAM, 8GB HBM
    - Logic Resources: 1,079,000 (Look-up Tables)

- **SW Development Kit**
  - Xilinx Vitis™ Unified Software Platform
    - Host program: C++ with OpenCL API, g++ compiler
    - FPGA kernels: OpenCL, v++ compiler
Final Goal

- Enhance the performance of MPI K-NN by reducing the amount of data transfer among MPI ranks using MEX
  - by using MEX as a communication buffer for MPI ranks in multi-node system

Development plans (2023~2024)

- Support multi-node systems
- Proprietary MEX API for the performance improvement
- Change PCIe to CCIX/CXL interface
- Optimized MVAPICH library for MEX


Thank You!