CHALLENGES AND GOALS

Research Challenges
- Design a low-level benchmark suite to analyze the efficiency of offloading collective communication patterns to SmartNICs.
- Examine simple algorithms for each pattern and empirically determine the number of DPU-based "workers" that would give optimal offload efficiency.
- Explore efficient/non-efficient algorithm designs to showcase what may happen if offload schemes are made inefficiently.

Research Goals
- Offload Efficiency: max reference time / max (pure_comm, compute) * 100
- Several assumptions made about runtime:
  - Block style hostfile ➔ Higher-numbered processes are on the DPU
  - Use of Multi-Program/Multi-Data mode in MPI libraries
- MPMD-Mode + Block Hostfile: Helps organize config file passed in at runtime

THE NEED FOR A NEW MICROBENCHMARK SUITE

- All the micro/benchmark suites that exist today are NOT DPU-aware (OMB, IMB, OpenHPCA, SMB, etc.) That is, a standard MPI library will not know whether a process is placed on a CPU or a DPU and run operations naively. Previous works have offloaded communication, computation, and Deep Learning to DPUs from the context of applications and libraries.
- With SmartNICs becoming more widespread, we need more ways of measuring their efficacy in the context of HPC and Datacenter environments.

RESEARCH MOTIVATION

NVIDIA’s BlueField DPU and others are becoming widespread in HPC clusters. Because of this, we need a DPU-aware micro-benchmark suite to determine how efficient they are in offloading communication operations. Previous research has designed ways to measure the offload processes placed on a DPU, but no efforts have been made along the micro-benchmark side.

EXPERIMENTS AND SETUP

HPC-AI Advisory Council Cluster – "Thor"
Partition
Running at 8-Nodes, 8-PPN on the host side, up to 64 workers total (8 WPN) on the DPU side, on messages ranging from 256KB to 4MB

DPU-Bench: Design and Implementation

- Why not MPI? Naively placing processes on a host server and a DPU will result in message progression being done on both pieces of hardware for nonblocking communication. Progress will become a bottleneck.
- Why IB-Verbs? RDMA semantics ➔ All operations can be issued from the DPU with no message progression being performed on the host server – making nonblocking communication through network primitives with the use of MPI for process management tool.

COLLECTIVES EXPLORLED

- Current Work: Linear algorithms for broadcast, reduce, and allgather
  - Allgather features an inefficient and an efficient design
  - Explore cyclic and block distribution of work and impact on load balancing and work distribution

FUTURE WORK

- Advanced algorithms
  - Generalize using UCX
  - Generalize to other programming models
  - Generalize to other SmartNICs

REFERENCES