

Challenges in Hardware Offloading of Collective Operations

Alex Margolin

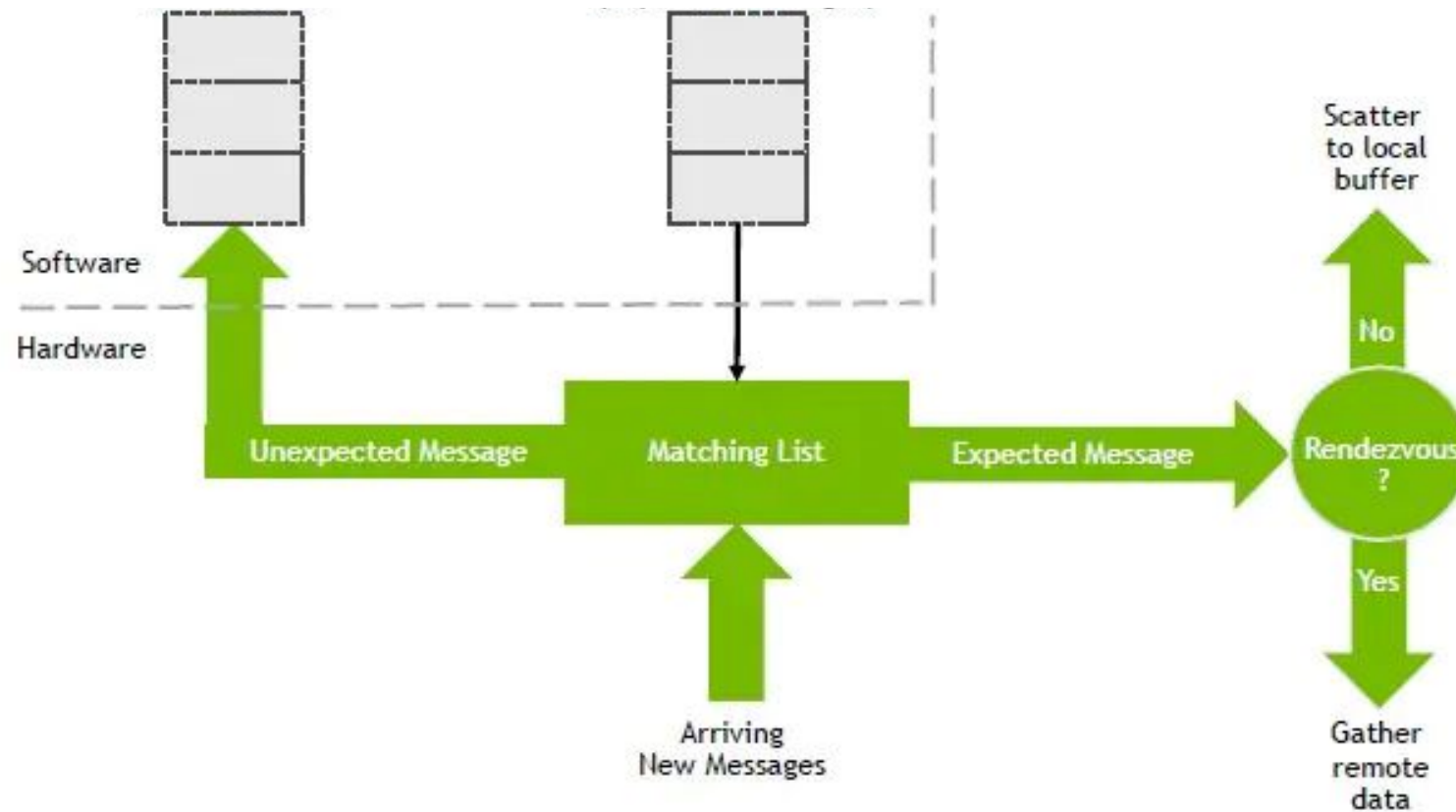
Ph.D. Candidate, Hebrew University of Jerusalem

Outline

- Approaches in collective offloading: Industry and Academia
- Challenges in offloading collective operations

Offloading P2P Communication

Example: *Hardware Tag-matching (Mellanox / NVIDIA)*



Offloading P2P Communication

Example: *Hardware Tag-matching (Mellanox /*

NVI

Benchmark	Nodes/PPN	Impact
AMG	192/6	Small Regression
HACC	128/6	No Impact
UMT	192/6	No Impact

Table 4: The impact of using hardware tag-matching on three applications.

Message Size	Base Latency	HW Tag Matching	Higher/(Lower) Latency (ns)
0	1.20	1.22	20
1	1.17	1.23	60
2	1.17	1.23	60
4	1.17	1.23	60
8	1.18	1.22	40
16	1.19	1.23	40
32	1.21	1.24	30
64	1.23	1.26	30
128	1.31	1.32	10
256	1.64	1.63	-10
512	1.76	1.70	-60
1024	2.00	1.83	-170
2048	2.88	2.67	-210
4096	3.52	3.44	-80
8192	5.00	4.69	-310

Table 2: OSU Latency Benchmark: Latency between two nodes with and without hardware tag-matching enabled. The fourth column is the difference between baseline and the offloaded, HW tag-matching. Negative values indicate HW TM has lower latency and positive numbers show higher latency. For messages 256 bytes to 8 KiB, HW TM reduces latency.

So far, no significant impact observed in key applications...

Offloading P2P Communication

Example: *Hardware Tag-matching (Mellanox / NVIDIA)*

Another example: *sPIN (ETH Zürich)*

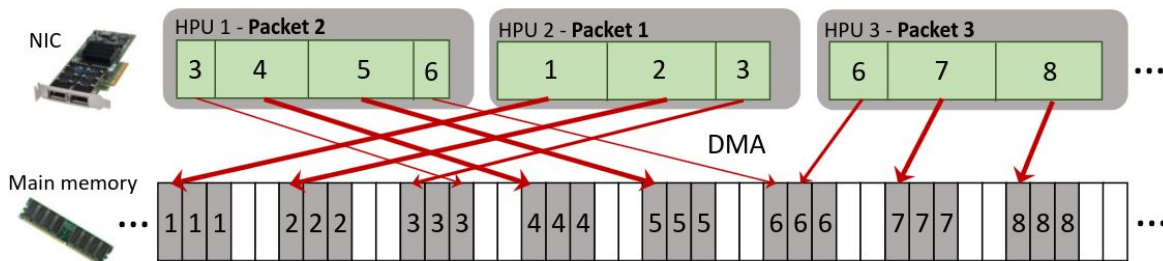


Figure 6: Processing vector datatypes in payload handlers

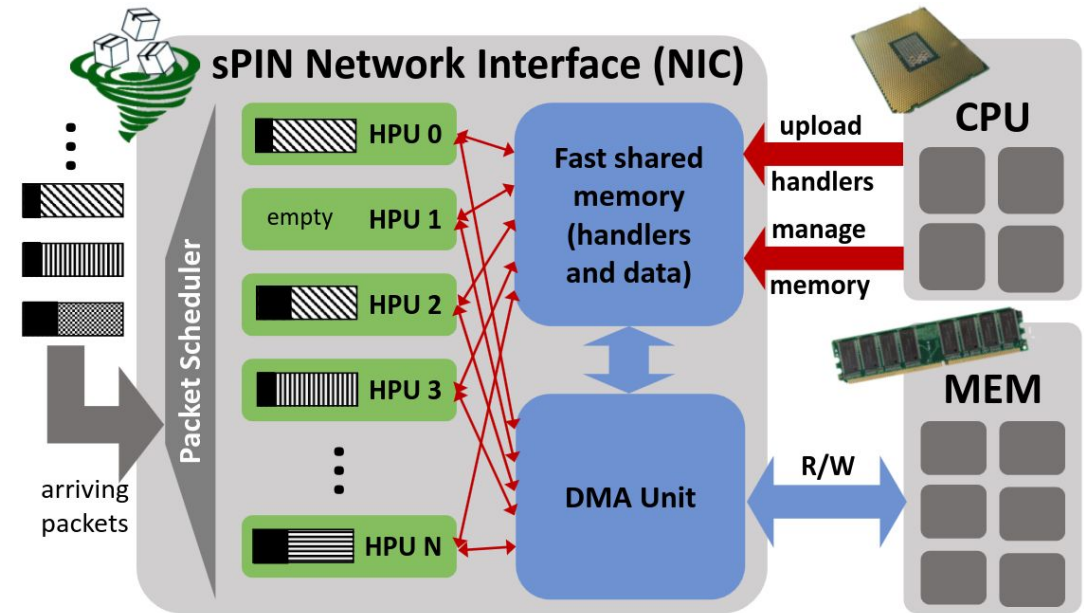
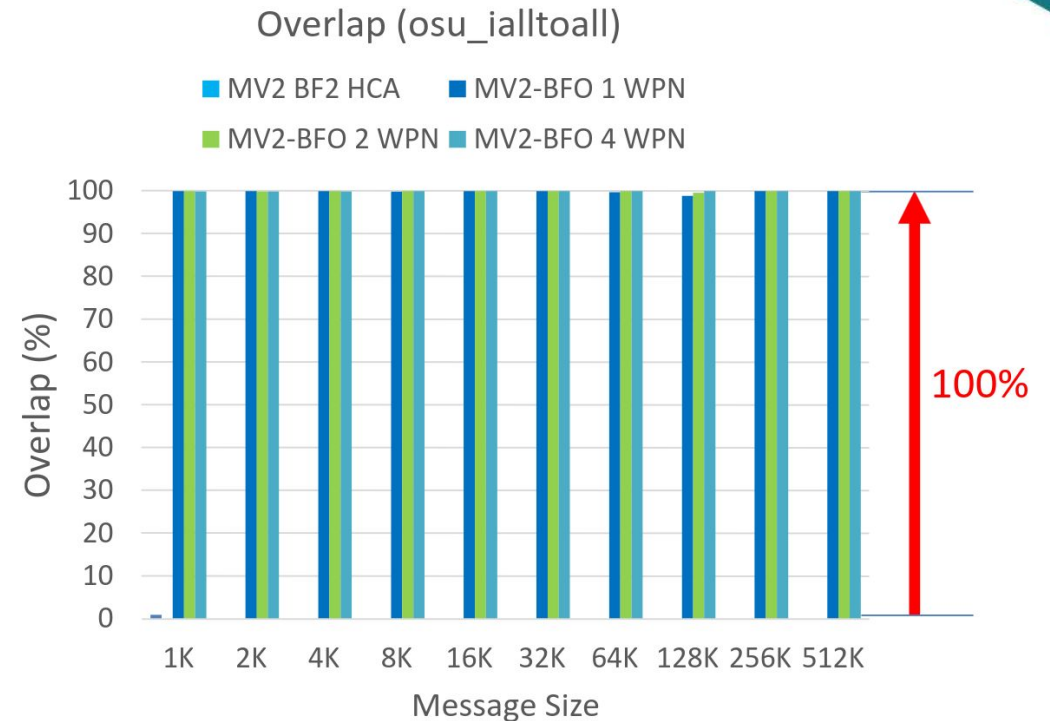
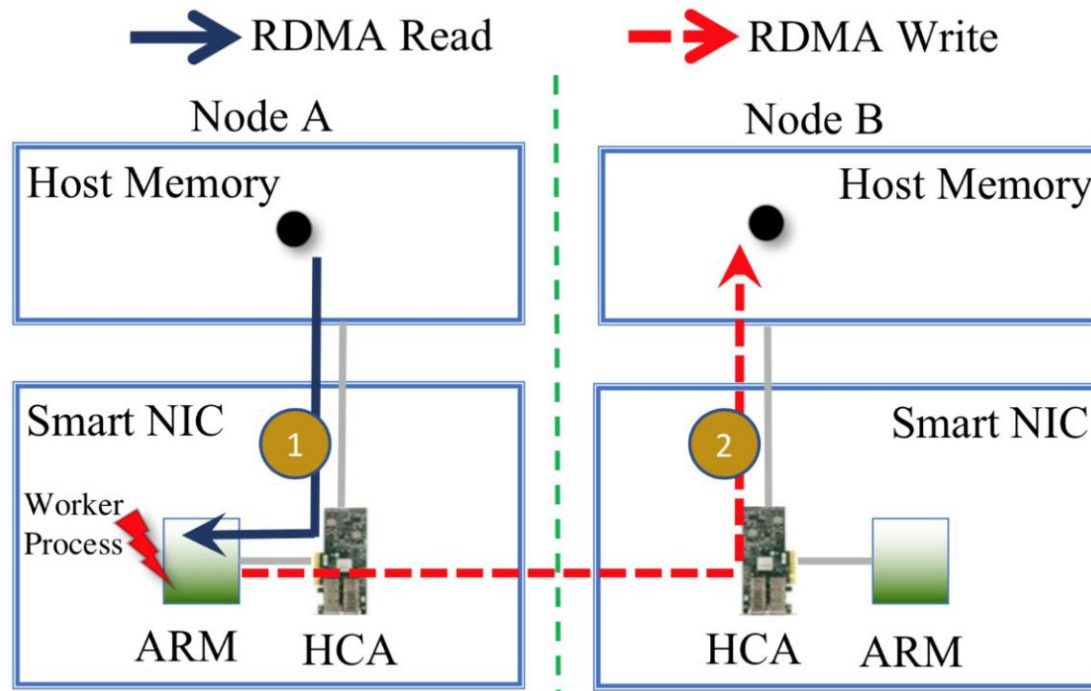


Figure 1: sPIN Architecture Overview

Offloading to Network Endpoints

Example: *Bluefield DPU (Mellanox / NVIDIA)*



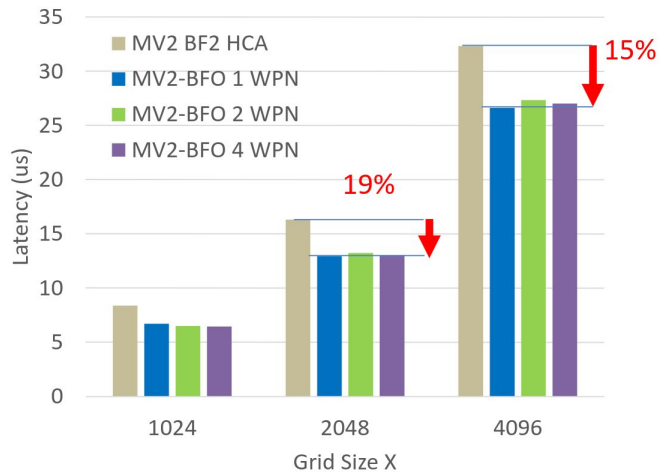
[3] *BluesMPI: Efficient MPI Non-blocking Alltoall offloading Designs on Modern BlueField Smart NICs*, by Bayatpour et al. (ISC '21)

[4] *Accelerating CPU-based Distributed DNN Training on Modern HPC Clusters using BlueField-2 DPUs*, by Jain et al. (HotI '21)

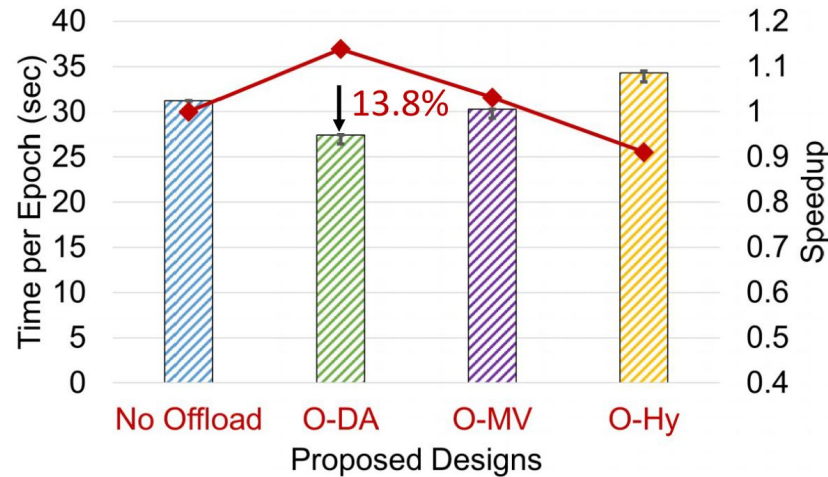
[5] *Large-Message Nonblocking MPI_iallgather and MPI_ibcast Offload via BlueField-2 DPU*, by Bayatpour et al. (HiPC '21)

Offloading to Network Endpoints

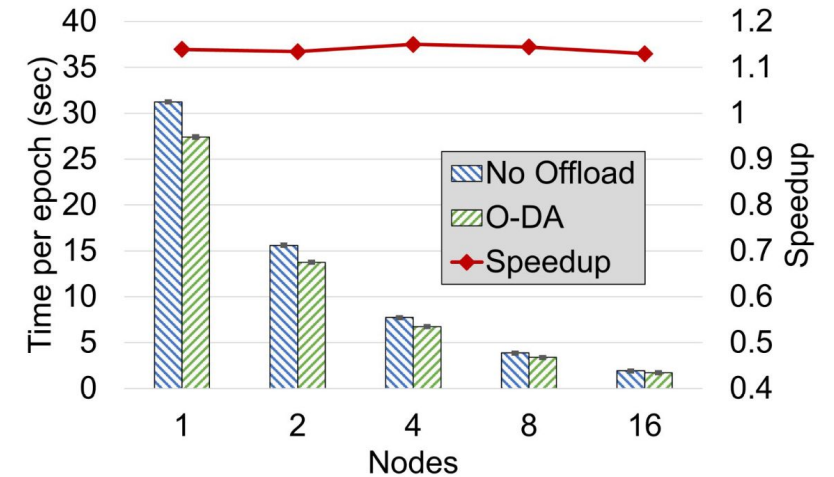
Example: *Bluefield DPU (Mellanox / NVIDIA)*



P3DFFT Application on 32 Nodes (16 PPN)



RESNET-20 Training on CIFAR-10 dataset - Single node (left) and multi-node (right)



[3] *BluesMPI: Efficient MPI Non-blocking Alltoall offloading Designs on Modern BlueField Smart NICs*, by Bayatpour et al. (ISC '21)

[4] *Accelerating CPU-based Distributed DNN Training on Modern HPC Clusters using BlueField-2 DPUs*, by Jain et al. (HotI '21)

[5] *Large-Message Nonblocking MPI_allgather and MPI_ibcast Offload via BlueField-2 DPU*, by Bayatpour et al. (HiPC '21)

Offloading to Network Endpoints

Example: *Bluefield SmartNIC (Mellanox / NVIDIA)*

Another example: *FPGA-based Offloads*



Fig. 1: A node of our cluster consisting of a Zedboard and an EthernetFMC daughter card.

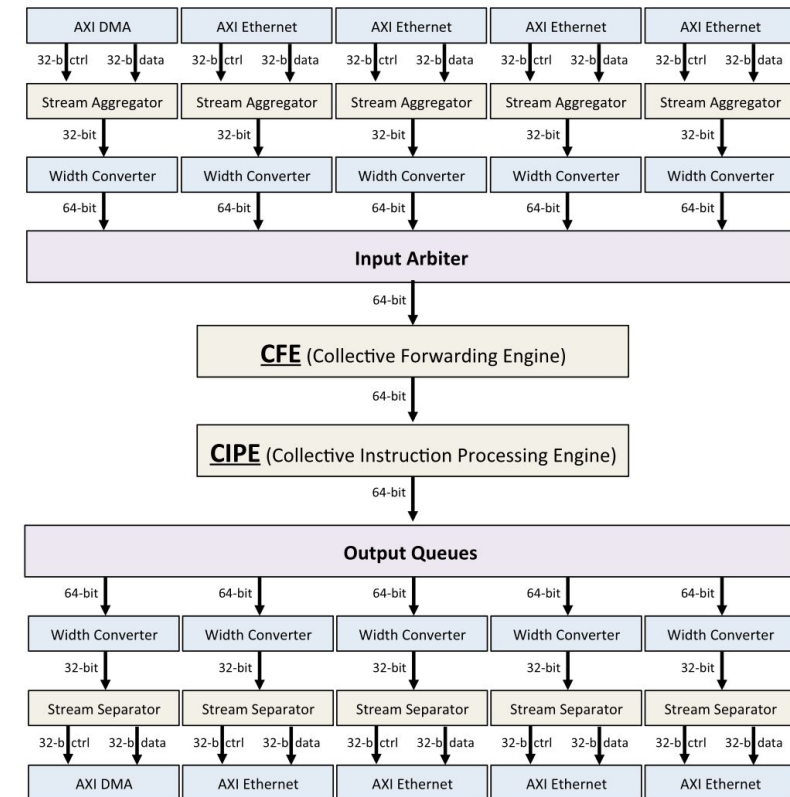
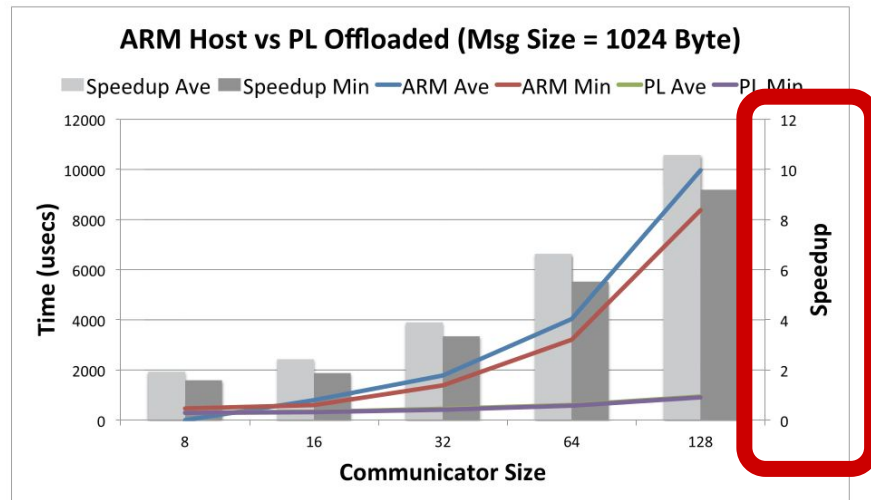


Fig. 3: High level architecture of collective operation offload design in programmable logic

Offloading to Network Endpoints

Example: *Bluefield SmartNIC (Mellanox / NVIDIA)*

Another example: *FPGA-based Offloads*



(c) 1024 byte message

Fig. 6: Microbenchmark comparison of *MPI_Allreduce* on *MPI_DOUBLE* with *MPI_SUM* operation running on the ARM host and PL offloaded version.

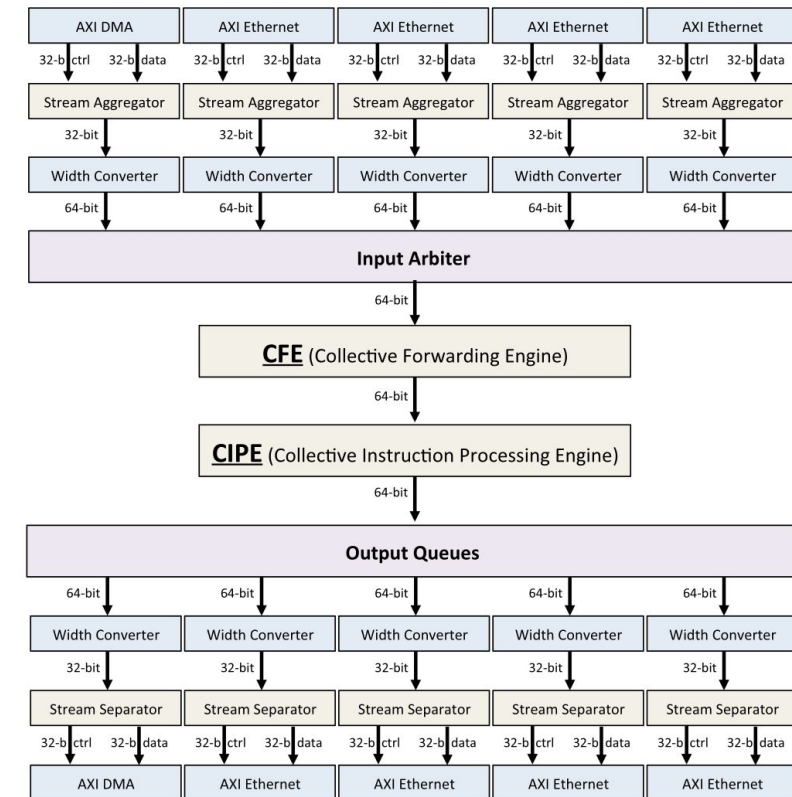


Fig. 3: High level architecture of collective operation offload design in programmable logic

Offloading to Network Endpoints

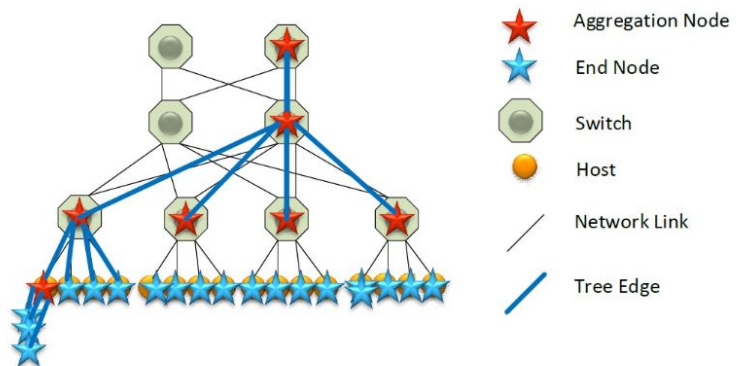
Example: *Bluefield SmartNIC (Mellanox / NVIDIA)*

Another example: *FPGA-based Offloads*

One more example: *COMET (Research project by Huawei - TRC)*

In-network Compute

Case study: **SHArP** Scalable Hierarchical Aggregation and Reduction Protocol (Mellanox / NVIDIA)




(b) Logical SHArP Tree. Note that in the SHArP abstraction an Aggregation Node may be hosted by an end-node.

Fig. 1: SHArP Tree example.

TABLE V: *MPI_Allreduce()* average latency (μ -seconds) on 128 hosts, one process per host. Comparison of pipelined SHArP-based algorithm with host-based algorithm.

Message Size [B]	SHArP based	Host Based	SHArP improvement factor
8	2.76	5.82	2.11
16	2.76	5.91	2.14
32	2.86	6.04	2.11
64	3.01	6.76	2.25
128	3.24	7.37	2.27
256	3.50	8.99	2.57
512	4.06	11.11	2.74
1024	5.49	18.04	3.29
2048	8.44	33.61	3.98
4096	14.48	46.93	3.24

In-network Compute

Case study:  Scalable Hierarchical Aggregation and Reduction Protocol (Mellanox / NVIDIA)

Another example: *FPGA-based Offloads*

Table 2: MPI_FPGA Speedups over OSU Benchmarks on BU SCC (128 byte messages)

MPI_FPGA Speedup Over OSU			
	32 ranks	64 ranks	128 ranks
Reduce	8.92	10.23	9.98
Allreduce	8.78	9.15	9.74
Bcast	9.23	9.21	9.45
Scatter	13.72	13.72	15.01
Gather	8.37	8.87	7.99
Allgather	5.86	7.34	7.15

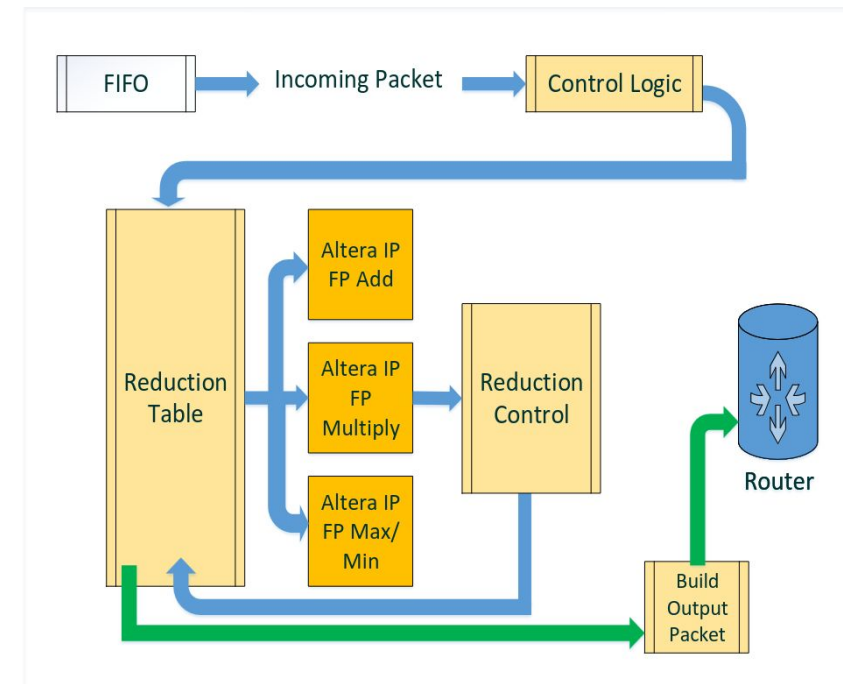


Figure 7: Flow of packets through the inner components of the reduction computation unit

Challenge #1: Programmability

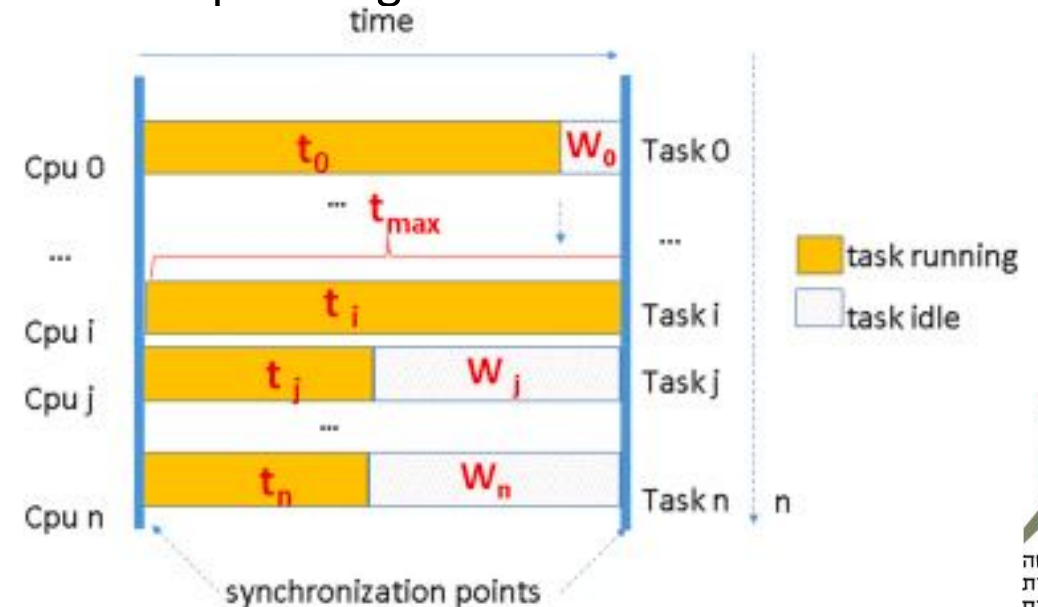
- MPI has a widely used collective operation API, but -
- No standard API for in-network offload definition/usage
 - NVIDIA DPUs use *DOCA SDK*
 - *OpenSNAPI*, a standardization effort doesn't seem to gain traction

What kind of API should a collective operation offload device support?

Challenge #2: Offload Resources

- Parallel programs often observe imbalance among processes
 - Imbalance in collective offloads means hardware resources are occupied
- More than one parallel program may share the offload resources
 - Avoiding resource allocation deadlocks requires careful planning
 - Fairness and QoS issues may arise...

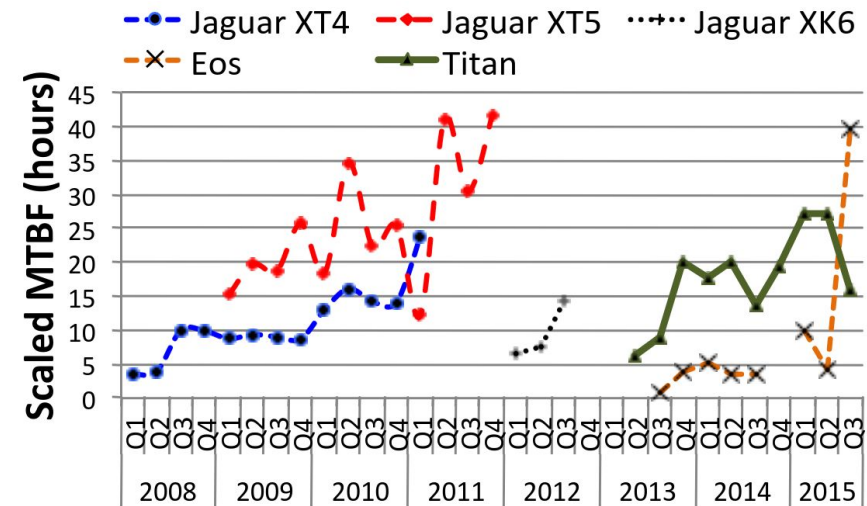
How much resources can be made available for each of the collective operations run in parallel?



Challenge #3: Fault-tolerance

- As systems grow in size/power/complexity - so do fault rates.
- A process might die without warning!
 - Leaving collectives “hanging”...

How to overcome a missing packet,
causing the entire collective to hang?



Scale-normalized MTBF of each system
over time (averaged quarterly)

Questions?