Battle of the BlueFields: An In-Depth Comparison of the BlueField-2 and BlueField-3 SmartNICs

***Abstract*—Over the past several years, Smart Network Interface Cards (NIC/SmartNICs) have rapidly evolved in popularity. In particular, NVIDIA's BlueField line of SmartNICs has been effective in a wide variety of uses: Offloading communication in High-Performance Computing applications (HPC), various stages of the Deep Learning (DL) pipeline, and is designed especially for Datacenter/virtualization uses. The BlueField-3 DPU was released at the end of 2022 as a follow-up to its widely accepted BlueField-2 predecessor, and this work will serve as an in-depth performance evaluation between the two to show a) a comparison of both SmartNICs' on-chip capabilities (memory bandwidth, compute speed, etc.), and b) their offload capabilities through several micro/benchmarks and applications. In single-DPU programs, we see up to 61% improvements in the latency of a memcpy operation and up to 82% bandwidth improvement in the use of the STREAM benchmark [8] on the BlueField-3. With the use of a DPU-aware MPI library [1], we observe over 30% improvement at the micro-benchmark level when comparing staging-based designs on both SmartNICs and up to nearly double that in the context of an application with staging-based designs. However, GVMI (Guest Virtual Machine ID) based designs contained in said library do not exceed 10% at the benchmark level and provide less than 2% benefits in applications because of its architecture-insensitive nature — that is, while CPU clock speed may impact the completion time of instructions, the performance of the GVMI-based designs in a DPU-aware MPI library will largely be unaffected by swapping the BlueField-2 for a BlueField-3.**

***Index Terms*—Datacenter Processing Units, BlueField-2, BlueField-3, SmartNIC, High-Performance Computing, Offload, Interconnects**

I. Introduction

Data processing units (DPUs) and other SmartNICs have rapidly grown in popularity over the past couple of years. The BlueField-2 DPU (BF2) [3], released at the end of 2020/start of 2021, was picked up by HPC researchers around the world and experimented on to determine how effective its components were beyond the capabilities that come with standard NICs. The BlueField-3 DPU (BF3) [12], featured at the end of 2022 and released at the beginning of 2023, comes equipped with up to double the bandwidth and compute power as well as substantially faster on-chip memory than that of its predecessor. This paper presents an in-depth evaluation and comparison of the compute capabilities of the BlueField-2 and BlueField-3 DPU[1] over various micro/benchmarks and applications.

This work was supported by Los Alamos National Laboratory/US Department of Defense, Contract #19537 and NSF grants #2007991 and #2018627.

[1]For the sake of simplicity, we interchange the phrases "DPU" and "SmartNIC" when discussing the BlueField series in this paper.

A. Motivation: SmartNICs are getting smarter

The BF2's release into the HPC and Datacenter communities provided not only a new NIC but one with previously unseen compute capabilities to aid in the work brought on by increasing HPC and DL workloads. Previous works (see Section II) have offloaded communication to the BF2 within the context of MPI libraries or DL applications, though with the further increased capabilities on the BF3, systems can arise where a user can offload more computation-intensive tasks in place of/in conjunction with communication.

B. Contributions

This paper makes the following contributions:

1) An in-depth comparison of the BF2 and BF3 DPUs on various micro/benchmarks in DPU-to-DPU communication.
2) Analysis of performance in micro/benchmarks and applications on host/DPU communications (host to BF2, host to BF3), both with and without a DPU-Aware MPI library.
3) A comparison of the BF3's capabilities when used on a system in which the host's memory and CPU speed may be outpaced by the SmartNIC, giving rise to systems that may have slower host CPUs and/or memory.

To the best of our knowledge, this is the first work that makes this kind of comparison between both SmartNICs and is the first work to examine the capabilities of the BlueField-3 DPU.

C. Paper Breakdown

The rest of this paper is broken down as follows: Section II will detail the background and overarching design of both the BF2 and BF3. Section III will break down our experiments and the results and inferences obtained from them. Section IV will showcase work related to SmartNICs and their use in the HPC ecosystem. Section V concludes our paper and introduces further thoughts on the future of SmartNICs and their uses in HPC and DL environments.

II. Background

In this section, we will compare and contrast the designs presented in the BlueField-2 and BlueField-3 SmartNICs. Note that while there are three overarching types of SmartNICs (ASIC-based, FPGA-based, and System-on-Chip-based (SoC)) [11], both the BlueField-2 and BlueField-3 are in the latter camp. Furthermore, while both DPUs come equipped with encryption and other features surrounding virtualization such

as Single-Root I/O Virtualization, they are beyond the scope of this paper and are not discussed here.

A. BlueField-2 and BlueField-3 SmartNICs

See Table I for a detailed breakdown of both the BlueField-2 (BF2) and BlueField-3 (BF3) SmartNICs. The BF3 effectively has nearly double the capabilities of the BF2, though Section III will further show that this may not always translate to improvements in running benchmarks and applications.

III. Experiments and Evaluation

This section showcases our experiments and the results and analysis obtained from them.

A. Experimental Setup

Our experimental testbed consists of 16 nodes each equipped with containing the "Thor" partition on the HPC-AI Advisory Council's HPC Center [4]: Dual-socket Intel Xeon 16-core CPUs (E5-2697A V4 @ 2.60 GHz), NVIDIA ConnectX-6 HDR100 100Gb/s InfiniBand Adapters, and 256GB DDR4 RDIMMs running at 2400 MHz (or 4800 MT/s). While all 32 of these nodes contain one BlueField-2 (BF2) SoC chip, only 16 of them are equipped with the BlueField-3 (BF3) SmartNICs. Note that these Intel CPUs have a smaller L1 and L2 cache size than the BF3 — a 32KB L1 dcache and icache (each), 256KB of L2 cache, and 40MB of L3 cache. The BF2 is connected via a single port of 100 Gb/s EDR InfiniBand, and the BF3 is connected via a single port of 200 Gb/s HDR InfiniBand.

B. Intra-DPU Experiments

Here, we detail some simple/small single-DPU experiments, starting with a single-process environment before scaling up/utilizing more resources.

1) Speed of a memory copy: The first experiment is a simple test to determine memory copy latency; one process performs ten back-to-back memory copies, where we take the average of these copy times ranging from 64 bytes to 1GB. In this test, we also make a comparison against the memory of this particular host. Figure 1 shows the increasing disparity between the BlueField-2 and 3 (BF2/3) alongside the improved runtime of the BF3's on-board memory. In the smallest messages tested, system variance shows little difference, save for the BF2 always having a non-zero copy latency. The difference between the host and BF2 (between 30 and 60%), and then the larger difference between the BF2 and the BF3 (up to 72%) gets shown once we enter the kilobyte range. Firstly, the BF3's L1 and L2 cache sizes dwarf those of the host. Secondly, once we spill into main memory, the memory clock speed becomes the dominating factor instead of the cache size. Compute nodes with faster memory and/or CPUs with larger caches may decrease or reverse the trends found in these results, though such trends still do not discount the improvements shown here.

2) The STREAM benchmark: Our second set of experiments examines intra-DPU performance. For this, we utilize the STREAM benchmark [8]. STREAM is intended to measure the bandwidth of main memory, and we do this with array sizes of 10 million and 100 million of type double. Our results are obtained from averaging ten back-to-back executions. The memory bandwidth of the BF3 allows for up to 3.3X improvement in single-threaded executions of STREAM, and increasing the number of threads allows the BF3 to obtain up to 5.5X improvement. While the BF3 has 16 cores, we keep the comparison from one to eight OpenMP threads when running tests on both the BF2/3 and the host, as shown by Figures 2a and 2b. Using 16 OMP threads on the BF3 gives no more than 5/6% improvement over eight threads, as the memory bandwidth on them gets saturated at this point.

When scaling the problem size up to a 100-million-element array, we note that the BF2 makes no improvements given that its memory bandwidth gets saturated when using two threads and see minor improvements for both the host and the BF3 when scaling up to eight threads.

C. Inter-DPU Experiments

We utilize the OSU Microbenchmark Suite (OMB) [13] for Inter-DPU experiments. In particular, we focus on MPI-based point-to-point. We utilize three point-to-point OMB MPI benchmarks: latency and uni/bi-directional bandwidth. The numbers shown in the following figures are the average of seven back-to-back executions. In particular, we examine the traditional range of message sizes for point-to-point HPC workloads (up to 4MB). Here, we purely discuss inter-DPU experiments without any comparison from the host. For these and other OMB-related experiments, we utilize the MVAPICH library developed and maintained by The Ohio State University [14].

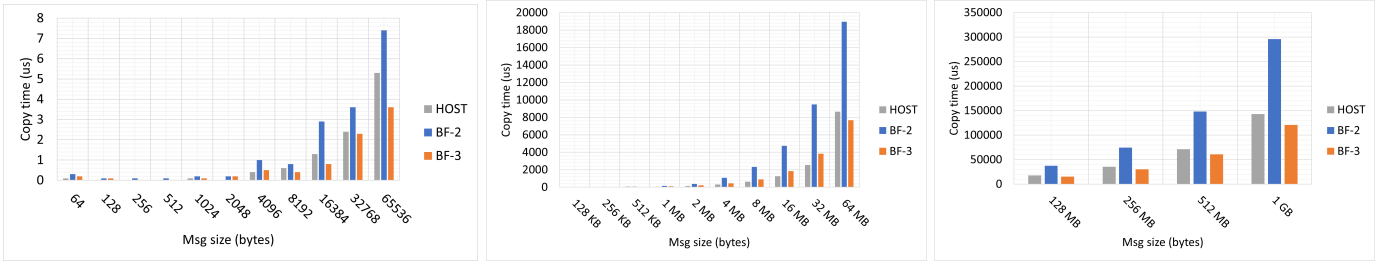
For point-to-point communications, we ensure that the right HCA is used when communicating messages — that is, BF2/3 DPUs communicate over their own adapter as opposed to traversing the bus to utilize a different HCA on the host nodes to which they are connected.

Figures 4 and 5 show the results of inter-DPU, point-to-point latency, where we observe up to 1.78X and 2.34X improvements between small (less than 16KB) and large messages, respectively. The majority of improvements the BF3 gains over the BF2 comes from information previously mentioned: firstly, we have a Smart/NIC featuring 2X the bandwidth as its predecessor; secondly, the increased L2 and L3 cache size will help with being able to hold larger messages in cache without the need for cache thrashing and/or eviction when storing smaller messages. Lastly, the higher clock speed of a single BF3 core will be able to do more than that of a BF2 core.

Figures 6 and 7 show unidirectional bandwidth results. We see up to 1.65X and 1.50X bandwidth improvement at small and large messages, respectively. Similarly, Figures 8 and 9 show up to 1.48X and 1.73X improvements in bidirectional bandwidth for small and large messages, respectively. The

TABLE I: Breakdown of the BlueField-2 and BlueField-3 SmartNICs ([10], [12])

Metric	BlueField-2	BlueField-3
Processor	ARM Cortex A-72 (2.0 GHz)	ARM Cortex A-78 (3.0 GHz)
Core Count	8	16
L1 Cache (per-core)	32KB D-cache, 48KB I-Cache	64KB for both I/D-Cache
L2 Cache (per-core)	1MB	512KB
L3 Cache (shared)	6 MB	16 MB
On-Chip RAM	16GB	32GB
Memory Controller Count	1	2
DRAM clock speed	1600 MT/s (DDR4)	5600 MT/s (DDR5)
Max Interconnect Speed	200 Gb/s	400 Gb/s

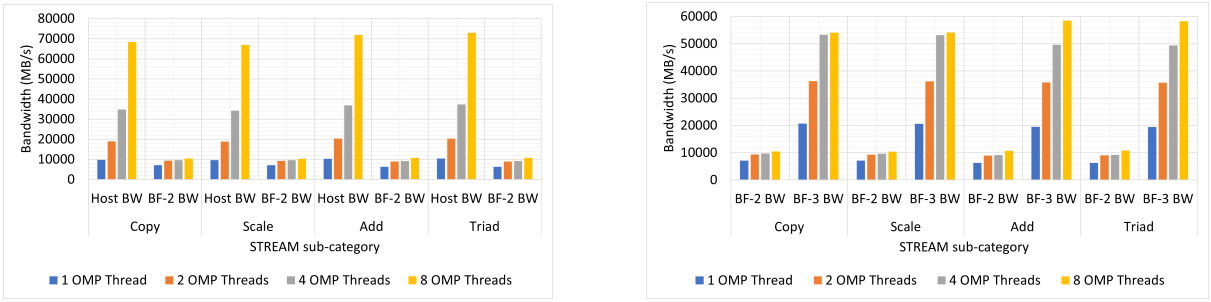


(a) Memcpy comparison on "small" messages up to 64KB

(b) Memcpy comparison on "medium" messages (64KB to 64MB)

(c) Memcpy comparison on "large" messages (128MB to 1GB)

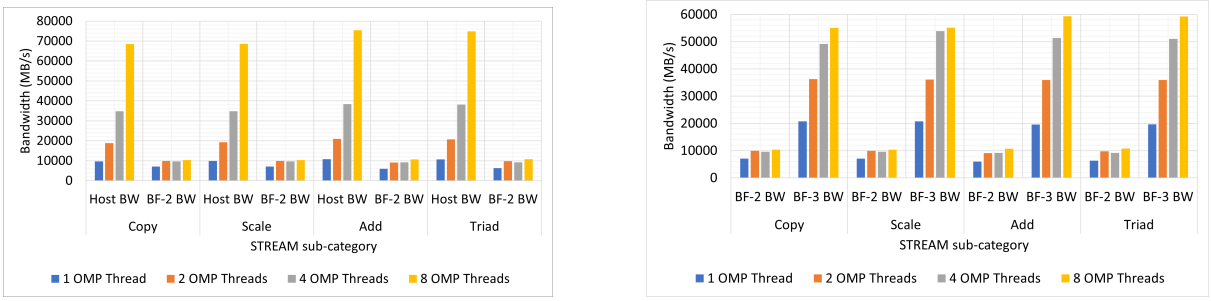
Fig. 1: Memcpy test to determine copy latency on host, BF2, and BF3. Being a single-process operation, the bottlenecks across all three environments are the clock speed of the CPUs involved, CPU cache sizes, and the clock speed of the memory controller(s).



(a) STREAM benchmark on an array of size 10-million (Host-versus-BF2)

(b) STREAM benchmark on an array of size 10-million (BF2-versus BF3)

Fig. 2: STREAM benchmark on varying OpenMP thread counts: Comparing a CPU to the BF2 and BF3 (10M-element arrays). Higher Bandwidth is Better



(a) STREAM benchmark on an array of size 100-million (Host-versus-BF2)

(b) STREAM benchmark on an array of size 100-million (BF2-versus BF3)

Fig. 3: STREAM benchmark on varying OpenMP thread counts: Comparing a CPU to the BF2 and BF3 (100M-element arrays). Higher Bandwidth is Better

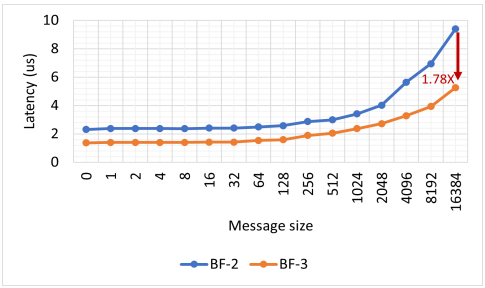


Fig. 4: Inter-DPU comparison of osu_latency (small msgs)

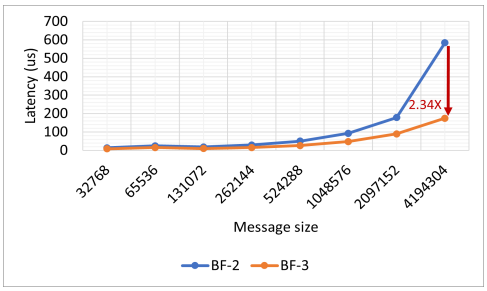


Fig. 5: Inter-DPU comparison of osu_latency (large msgs)

BF2 loses bandwidth at 4MB due to the memory requirements spilling into the L3 cache.

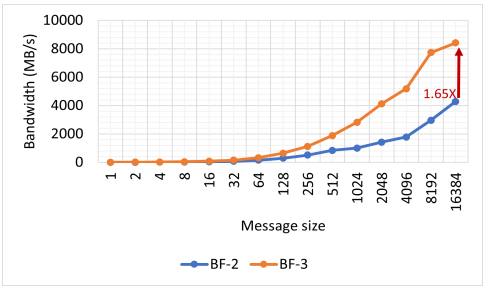


Fig. 6: Inter-DPU comparison of osu_bw (small msgs)

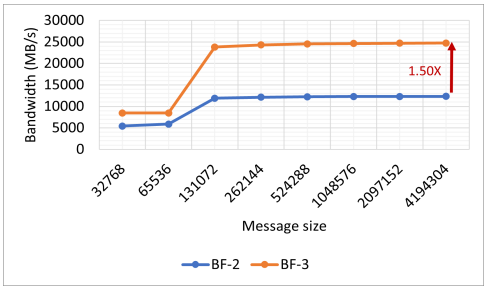


Fig. 7: Inter-DPU comparison of osu_bw (large msgs)

D. Host/DPU Experiments

We replicate the experiments performed in the previous section here, except with naive Host-DPU communication instead. The extra difference in running benchmarks and applications on CPU/DPU configurations is the need to use an MPI

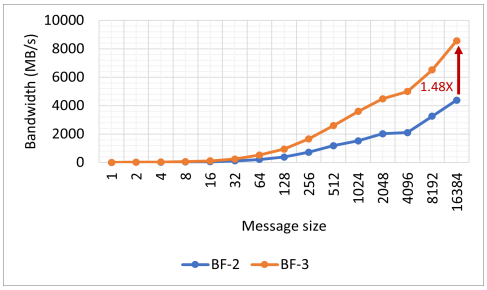


Fig. 8: Inter-DPU comparison of osu_bibw (small msgs)

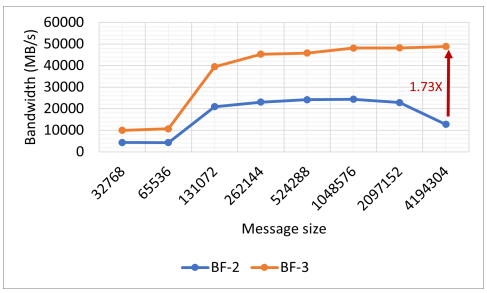


Fig. 9: Inter-DPU comparison of osu_bibw (large msgs)

library's "MPMD" mode[2] to run these experiments; the use of a configuration file or entering commands at the command line tells the MPI runtime to run both executables simultaneously, treating them both as if they were one large MPI program. In addition, we also use the IB-Verbs-based [7] DPU-Bench [9] benchmark suite for showing the offload efficiency of various collective operations onto each of the DPUs. DPU-Bench explores the offload efficiency of SmartNICS by presenting microbenchmarks to directly determine how efficient it is to offload algorithms for various communication patterns (one-to-all, all-to-one, all-to-all) across different message sizes and numbers of worker processes placed on the DPU. The higher the number on a scale of 1-100, the easier/more efficient a given configuration and message size can be offloaded to the DPU via naive staging.

Like in the previous subsection, these benchmark numbers were obtained after averaging seven back-to-back executions of each OMB benchmark. The DPU-Bench numbers were obtained after averaging three back-to-back executions of each benchmark on various numbers of worker processes placed onto each of the DPUs (these are labeled as "workers per node", or WPN).

1) OMB Results: In Figures 10 and 11, we see that naive utilization of the BF3 in place of the BF2 gives some performance benefits – up to 1.67X across small messages, but no more than 13% at large messages. Not only are we dealing with a heterogeneous environment in these cases, but the disparity in resources between the host and the DPU leads to less of a performance improvement when swapping the BF2 out for the BF3. A similar trend is found in the bi-

[2]See https://www.intel.com/content/www/us/en/develop/documentation/mpi-developer-guide-linux/top/running-applications/mpmd-launch-mode.html

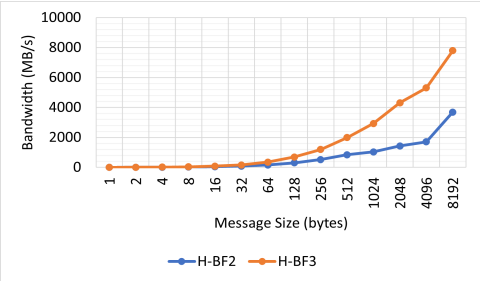


Fig. 10: Host-DPU comparison of osu_bw (small msgs)

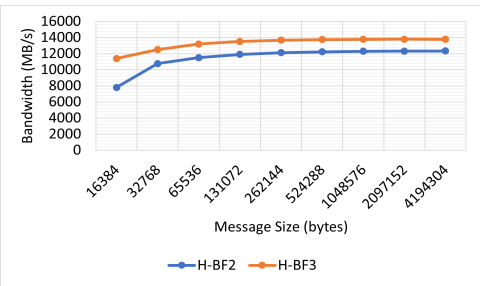


Fig. 11: Host-DPU comparison of osu_bw (large msgs)

directional bandwidth results shown in Figures 12 and 13, with the additional fact that the smaller caches on the BF2 degrade performance sooner than the BF3.

Host-DPU latency (Figures 14 and 15) behaves similarly to DPU-DPU latency (Figures 4 and 5); increased bandwidth and a larger cache size allow the BF3 to outpace the BF2 in message latency when receiving from the host.

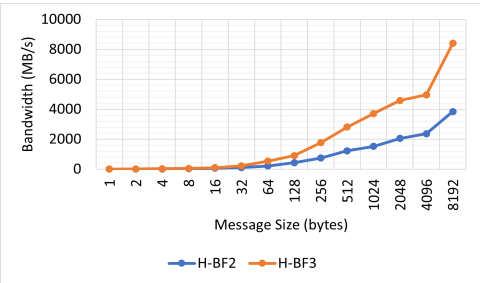


Fig. 12: Host-DPU comparison of osu_bibw (small msgs)

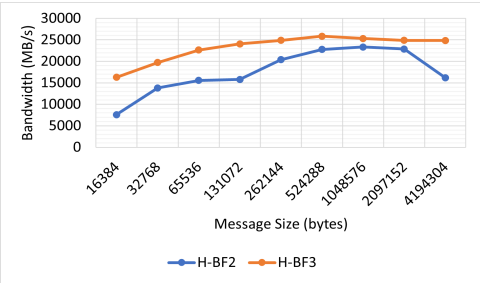


Fig. 13: Host-DPU comparison of osu_bibw (large msgs)

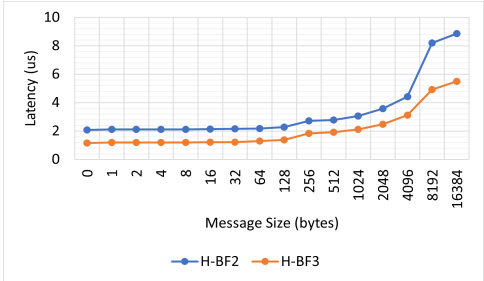


Fig. 14: Host-DPU comparison of osu_latency (small msgs)

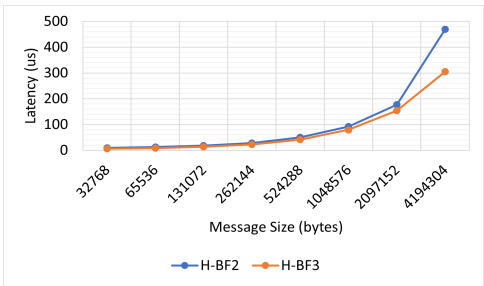


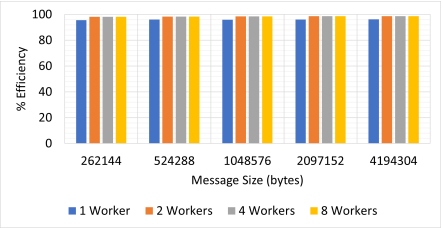
Fig. 15: Host-DPU comparison of osu_latency (large msgs)

2) DPU-Bench: In this section, we compare the offload efficiency of the BF2 and BF3 among the "Cyclic" work assignments mentioned in [9]. As mentioned in that work, the offload efficiency calculation is $\frac{ref\_time}{max(compute,pure\_comm)} * 100$. We run these experiments at a scale of eight nodes, eight processes per node (PPN), and varying numbers of workers, from one worker total to eight worker processes (one WPN), which get placed on each of the BF2's (or BF3) cores.

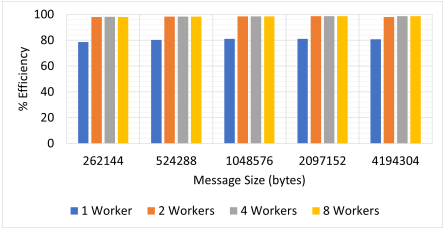
We start by displaying the offload efficiency when using anywhere between one worker total, up to one WPN in Figures 16 and 17. The cyclic work assignment for broadcast and gather-like communication patterns give similar benefits for both SmartNICs, though we note that we achieve a higher level of offload efficiency on the BF3 with even just one worker, especially in "Cyclic-Gather's" case. What is most impressive is that the communication time in the "Multi-Root-Cyclic-Allgather" gets exceeded by the compute time as we hit one WPN, leading to hitting 100% offload efficiency — this means that, while efficient communication offload mechanisms are preferred, it is easier to more naively offload communication patterns with the BF3's advanced hardware.

Similar trends exist for DPU-Bench as shown in Figures 18 and 19. In "Multi-Root-Cyclic-Allgather", this unusual work distribution is a massive point of contention as the degradations shown in the runs on the BF2 DPU get amplified in the runs on the BF3. Several factors could contribute to this, such as congestion on either the PCIe bus or the InfiniBand network itself.

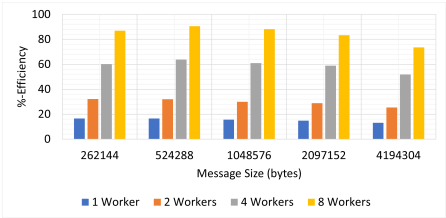
3) In the context of Applications w/ a DPU-Aware MPI library: Here, we examine the use of a DPU-aware MPI library on an application and compare the results from the BF2 and BF3 SmartNICs. We obtained a license from X-



(a) DPU-Bench's "Cyclic-Bcast" between the host and BF2 (8 nodes, 8 PPN, up to 1 WPN)

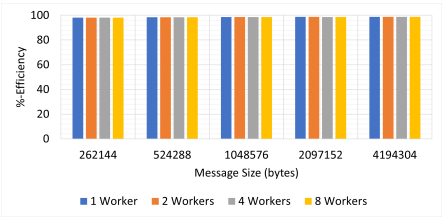


(b) DPU-Bench's "Cyclic-Gather" between the host and BF2 (8 nodes, 8 PPN, up to 1 WPN)

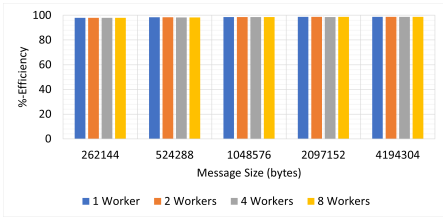


(c) DPU-Bench's "Multi-Root-Cyclic-Allgather" between the host and BF2 (8 nodes, 8 PPN, up to 1 WPN)

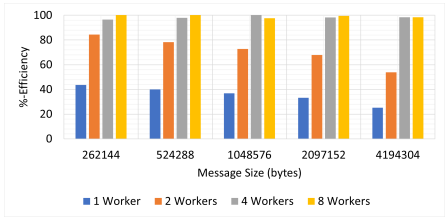
Fig. 16: DPU-Bench results between the host and BF2 DPU at an 8-node, 8 PPN scale, up to 1 WPN



(a) DPU-Bench's "Cyclic-Bcast" between the host and BF3 (8 nodes, 8 PPN, up to 1 WPN)

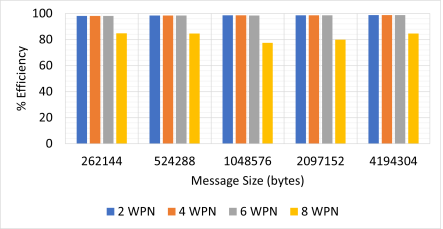


(b) DPU-Bench's "Cyclic-Gather" between the host and BF3 (8 nodes, 8 PPN, up to 1 WPN)

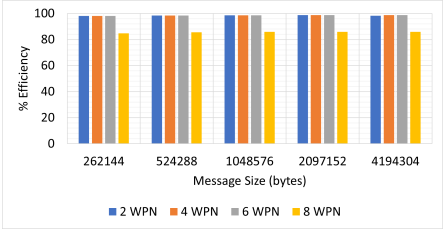


(c) DPU-Bench's "Multi-Root-Cyclic-Allgather" between the host and BF3 (8 nodes, 8 PPN, up to 1 WPN)

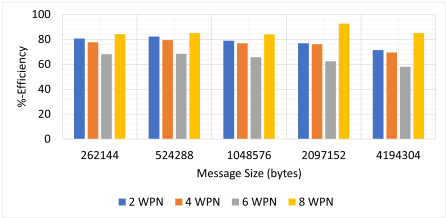
Fig. 17: DPU-Bench results between the host and BF3 DPU at an 8-node, 8 PPN scale, up to 1 WPN



(a) DPU-Bench's "Cyclic-Bcast" between the host and BF2 (8 nodes, 8 PPN, from 2 WPN to 8 WPN)

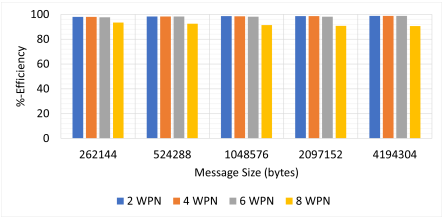


(b) DPU-Bench's "Cyclic-Gather" between the host and BF2 (8 nodes, 8 PPN, from 2 WPN to 8 WPN)

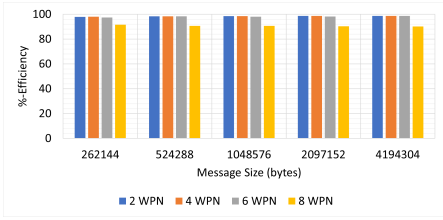


(c) DPU-Bench's "Multi-Root-Cyclic-Allgather" between the host and BF2 (8 nodes, 8 PPN, from 2 WPN to 8 WPN)

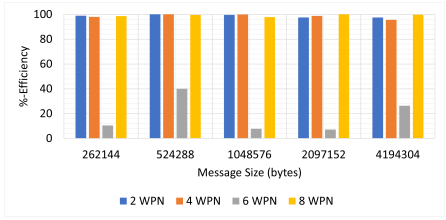
Fig. 18: DPU-Bench results between the host and BF2 DPU at an 8-node 8 PPN scale, from 2 WPN to 8 WPN



(a) DPU-Bench's "Cyclic-Bcast" between the host and BF3 (8 nodes, 8 PPN, from 2 WPN to 8 WPN)



(b) DPU-Bench's "Cyclic-Gather" between the host and BF2 (8 nodes, 8 PPN, from 2 WPN to 8 WPN)



(c) DPU-Bench's "Multi-Root-Cyclic-Allgather" between the host and BF3 (8 nodes, 8 PPN, from 2 WPN to 8 WPN)

Fig. 19: DPU-Bench results between the host and BF3 DPU at an 8-node, 8 PPN scale, from 2 WPN to 8 WPN

ScaleSolutions [1] to use their MVAPICH2-DPU library and with it, we show two sets of results: 1) the improvement BF3 DPUs give over BF2s in osu_ialltoall (Figures 20, 21, and 22 for sixteen-node results), and 2) through the use of a modified P3DFFT [15] to utilize nonblocking alltoall calls at 8 and 16 nodes with various PPN (Figures 23 and 24). In addition to staging-level results, we also show the use of GVMI-based designs within MVAPICH2-DPU, where GVMI (Guest Virtual Machine ID) exists as firmware on the DPU which exposes memory regions from the host to the DPU.

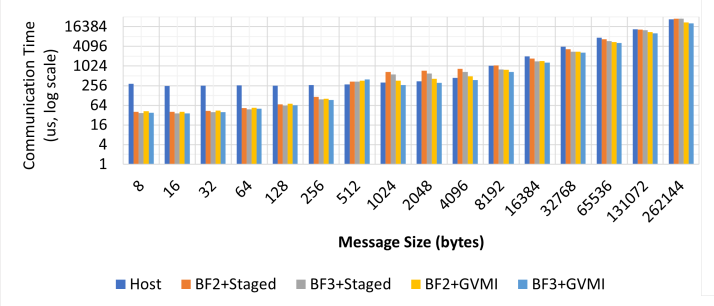


Fig. 20: osu_ialltoall with MVAPICH2-DPU (16 nodes, 8 PPN on host)

Our osu_ialltoall results are averaged over four back-to-back executions with a comparison to pure-host runtime results. As the message size increases, staging overhead causes degradations on both SmartNICS, even with the BF3 having faster cores than the host servers used. For small and medium message sizes, system variances can dictate results that would lead to at most a 5% degradation in the BF3's offload capabilities, as observed at 512 bytes in Figure 21 and 256 bytes in Figure 22. While Staging shows substantial benefits between the two DPU models, GVMI does not show more than 10% benefits at the benchmark level thanks to its architecture-insensitive nature.

A mixture of large messages and resource contention in using all of the BF3's cores may result in staging-based designs performing worse on the BF3 than on the BF2. In the same figures, larger message sizes show up to 10% degradation on the BF3. While not shown in the paper, we also see a 15% degradation at 16 Nodes/4 PPN.

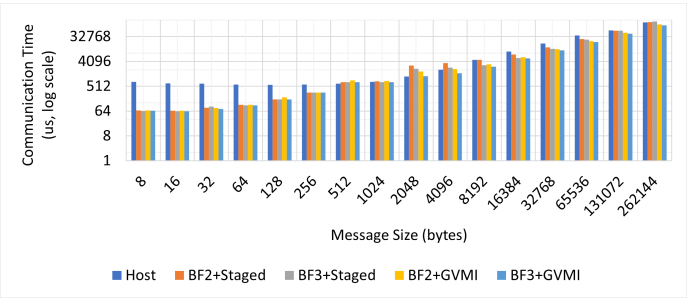


Fig. 21: osu_ialltoall with MVAPICH2-DPU (16 nodes, 16 PPN on host)

P3DFFT's process mapping internally used by its sample programs makes an impact on performance, so we try to mitigate any degradations by keeping the X and Y dimensions of the process mapping as (Node-count x PPN) for these our runs. This partially explains the lack of degradation/improvement seen at the smallest scale (8 Nodes/8 PPN) when using a 1920 x 1920 x 1920 mesh in Figure 23. Given the real-life compute shown here, GVMI shows benefits over staging-based designs, but again, its architecture-insensitive nature shows little to no benefits when comparing the BF2 against the BF3.

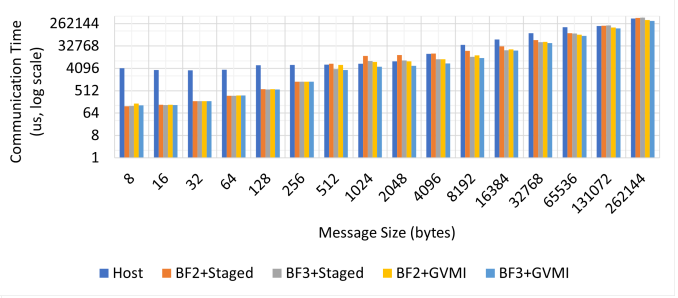


Fig. 22: osu_ialltoall with MVAPICH2-DPU (16 nodes, 32 PPN on host)

In this setup, we note that the pure-host execution does better than staging designs on the BF2 at a small scale (up to 2X against BF2), and better than the BF3 at a larger scale (up to 2X against BF3), though still worse than GVMI-based designs in both scales. Further analyses (profiling and tuning) with larger and smaller problem sizes/scales are needed along this line to show the efficacy of even naive staging for larger scales on both chips.

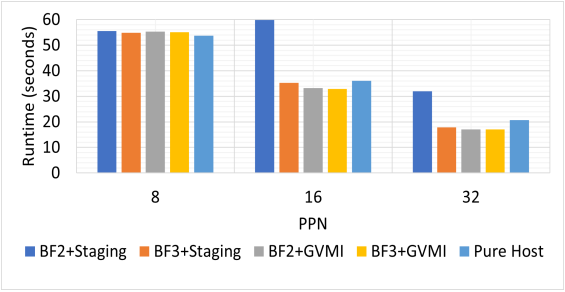


Fig. 23: P3DFFT using ialltoall at 8 nodes, various PPN

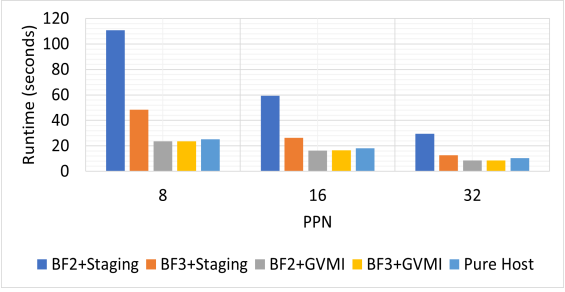


Fig. 24: P3DFFT using ialltoall at 16 nodes, various PPN

IV. Related Work

This section details some of the vast related work to current designs and research that have either used or examined DPUs in various designs/environments.

A. MPI-based designs and Benchmarks

The authors of [2] made the first proposal to use DPUs in the context of MPI libraries. This work employed an efficient staging design in the context of MPI_Ialltoall. [16] extended

this idea to MPI_Ibcast and MPI_Igather patterns. [9] explores preliminary designs of a benchmark suite to determine how efficient it is to offload different communication patterns to a DPU.

B. Applications in the HPC/DL communities

The authors of [6] performed a case study on MiniMD after modifying its code to recognize the BF2 and achieve a 20% increase in runtime with no loss in simulation accuracy. [5] showed how to enhance DL training through the use of DPUs, where various stages of DL training were placed on the BF2 to performance gains on three different models.

C. Applications outside of HPC

The authors of [17] utilized the BF2 for DBMS-based operations (Database Management Systems); they show how throughput gets affected when offloading portions of the B-Tree representing the DBMS. While most of the aforementioned MPI-related works mention the use of offloading communication, this and [6] make use of the DPU for offloading computation — a less trivial task on the BF2.

It should be noted that given the results from Section III, most of the efficient designs made for the BF2 will not only be applicable to the BF3 but could be improved to further take advantage of the enhanced hardware on later DPUs.

V. Conclusion and Future Work

In this paper, we have presented a comprehensive comparison of the performance of the BlueField-2 and BlueField-3 SmartNICs in various capacities: intra-DPU, both with simple, single and multi-threaded benchmarks, inter-DPU with industry-standard micro-benchmarks, and in the context of working with a host server (micro-benchmarks and applications) to judge the ability to receive offloaded communication data. We have systematically analyzed how much more powerful the BlueField-3 DPU is compared to its predecessor to the best of our abilities. We aim to design more intelligent offloading schemes to take better advantage of the BlueField-3 and other DPUs. With systems that have multiple SmartNICs/SmartNIC generations per node, or multiple generations of them such as the experimental cluster we used for this paper, it will be interesting to see if attempting to offload to multiple SmartNICs on a single node will result in some performance degradation or even further improvement given the disparity between both BlueField DPUs. We plan to expand our work to include potential offloadable applications potentially using OpenSHMEM and OpenMP.

VI. Acknowledgements

We would like to thank the HPC-AI Advisory Council for allowing us access to their resources. We would also like to thank X-ScaleSolutions for granting us a license to utilize their MVAPICH2-DPU software to efficiently experiment and examine hardware trends at the application level with a DPU-Aware MPI library. We would like to acknowledge funding from Los Alamos National Laboratory, the Department of Defense, and the National Science Foundation.

References

[1] X-ScaleSolutions. https://x-scalesolutions.com/.

[2] Mohammadreza Bayatpour, Nick Sarkauskas, Hari Subramoni, Jahanzeb Maqbool Hashmi, and Dhabaleswar K. Panda. BluesMPI: Efficient MPI Non-blocking Alltoall Offloading Designs on Modern BlueField Smart NICs. In Bradford L. Chamberlain, Ana-Lucia Varbanescu, Hatem Ltaief, and Piotr Luszczek, editors, *High Performance Computing*, pages 18–37, Cham, 2021. Springer International Publishing.

[3] Idan Burstein. Nvidia Data Center Processing Unit (DPU) Architecture. In *2021 IEEE Hot Chips 33 Symposium (HCS)*, pages 1–20, 2021.

[4] HPCAC, 2022.

[5] Arpan Jain, Nawras Alnaasan, Aamir Shafi, Hari Subramoni, and Dhabaleswar K Panda. Accelerating CPU-based Distributed DNN Training on Modern HPC Clusters using BlueField-2 DPUs. In *2021 IEEE Symposium on High-Performance Interconnects (HOTI)*, pages 17–24, 2021.

[6] S. Karamati, C. Hughes, K. Hemmert, R. E. Grant, W. Schonbein, S. Levy, T. M. Conte, J. Young, and R. W. Vuduc. "Smarter" NICs for faster molecular dynamics: a case study. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 583–594, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.

[7] Patrick MacArthur, Qian Liu, Robert D. Russell, Fabrice Mizero, Malathi Veeraraghavan, and John M. Dennis. An integrated tutorial on infiniband, verbs, and mpi. *IEEE Communications Surveys & Tutorials*, 19(4):2894–2926, 2017.

[8] John D. McCalpin. Memory Bandwidth and Machine Balance in Current High Performance Computers. *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pages 19–25, December 1995.

[9] Benjamin Michalowicz, Kaushik Kandadi Suresh, Hari Subramoni, Dhabaleswar Panda, and Steve Poole. Dpu-bench: A micro-benchmark suite to measure offload efficiency of smartnics. In *Practice and Experience in Advanced Research Computing*, PEARC '23, page 94–101, New York, NY, USA, 2023. Association for Computing Machinery.

[10] NVIDIA. NVIDIA BlueField-2 DPU Data Sheet, 2020.

[11] NVIDIA. Blog Post: Choosing the best DPU-based SmartNIC, 2021.

[12] NVIDIA. NVIDIA BlueField-3 DPU Data Sheet, 2022.

[13] OSU Micro-benchmarks. http://mvapich.cse.ohio-state.edu/benchmarks/.

[14] Dhabaleswar Kumar Panda, Hari Subramoni, Ching-Hsiang Chu, and Mohammadreza Bayatpour. The MVAPICH project: Transforming research into high-performance MPI library for HPC community. *Journal of Computational Science*, 52:101208, 2021. Case Studies in Translational Computer Science.

[15] Dmitry Pekurovsky. P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions. *SIAM Journal on Scientific Computing*, 34(4):C192–C209, 2012.

[16] Nick Sarkauskas, Mohammadreza Bayatpour, Tu Tran, Bharath Ramesh, Hari Subramoni, and Dhabaleswar K. Panda. Large-Message Nonblocking MPI_Iallgather and MPI Ibcast Offload via BlueField-2 DPU. In *2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, pages 388–393, 2021.

[17] Lasse Thostrup, Daniel Failing, Tobias Ziegler, and Carsten Binnig. A DBMS-centric Evaluation of BlueField DPUs on Fast Networks. May 2022. 13th International Workshop on Accelerating Analytics and Data Management Systems Using Modern Processor and Storage Architectures Event Title: 48th International Conference on Very Large Databases.