



Cross-Layer Data-Integrity Qualification of GPU-Direct Storage Paths Using NVMe-oF and DPU Offload

Gagan Vyas¹

¹ Member of Technical Staff, Pure Storage, Santa Clara, California, USA
Corresponding Author: gaganvyas@gmail.com

ARTICLE HISTORY

Received Dec, 2025

Revised Month Dec, 2025

Accepted Dec, 2025

ABSTRACT

GPUDirect Storage over NVMe over Fabrics (NVMe-oF), more often now combined with data processing unit (DPU) offload, has reduced the latency from flash to accelerator memory but eliminated the host-side copy at which payloads were conventionally sifted, and silent data corruption (SDC) is a documented problem at fleet scale. It was desired to find a cross-layer data-integrity qualification scheme for these paths. We performed a systematic review of 17 publications (3 journal articles, 7 conference or book-series papers and 7 preprints). The evidence was structured with nine equations that describe residual corruption probability, digest coverage, verification overhead, offload overlap and test-pass sizing and was categorised by function (path layer) and offloading. There were five path layers related to fault classes, detection mechanisms, qualification tests. NVMe-oF was reported to increase the access time by 11.7 μ s, while iSCSI decreased key-value throughput by 40%. Offload gains of 15% in training time and 2.8 times in throughput were achieved without reporting any overhead for verification. In an illustrative model, transport-only digests left about 977 expected undetected events per exabyte, while end-to-end coverage left 8.0×10^{-7} expected undetected events per exabyte. The level of integrity is a path-level property and needs to be accompanied by overheads for integrity verification along with offload efficiency. The review provides a multi-layer qualification matrix and a quantitative model that are still valid in case of migrating storage functions to the DPU.

Keywords: Data Integrity; Data Processing Unit; Ebpfi; GPUDirect Storage; NVMe over Fabrics; Qualification; Silent Data Corruption; Storage Disaggregation; Systematic Review

INTRODUCTION

Large models' training and analytics pipelines increasingly rely on continuous data streaming from connected flash to accelerator memory. Direct memory access (DMA) path from storage to graphics processing unit (GPU) memory, which bypasses the host bounce buffer (Direct memory access (DMA) storage to graphics processing unit (GPU) memory) and NVMe over Fabrics (NVMe-oF) — extends the Non-Volatile Memory Express (NVMe) command set over remote fabrics. This path can be implemented in either end and used to offload the host central processing unit (CPU) to a data processing unit (DPU) that includes a number of network interfaces, programmable Arm processors and fixed-function engines. Every offload, however, adds another place where payload bytes are staged, manipulated or confirmed. The accuracy of data that is delivered is thus not characteristic of the individual component, but of the entire path.

Errors may not be reported and a level of correctness cannot be inferred on this scale. Silent data corruption (SDC) is a malfunction of a hardware circuit with no check logic, causing the circuit to generate an incorrect result without any indication in system logs. These can switch a single bit or

cause the wrong instruction to be executed, and are exacerbated by datapath variation, temperature changes, and device age. Two fleet-scale screening strategies were compared, out-of-production screening (Fleetscanner) and in-production testing (Ripple) [1] on the basis of over three years of production experience. The contrast is directly relevant to the storage to GPU paths, as all of the steps along a path are silicon on which "silent faults" can arise, and a good verdict at one step cannot certify another step.

Supervising for more than 18 months on hundreds of thousands of machines led to the identification of hundreds of processors that exhibited silent corruption, suggesting that silent corruption is not a rare occurrence on a single machine, but rather a problem on many machines and across hardware generation [2]. That work found that hardware resilience, production detection and fault-tolerant software are all needed for mitigation. With this background in mind, the present systematic review aims to combine the evidence on NVMe-oF, GPU-direct storage, and DPU offload from 17 sources; to define a cross-layer qualification framework where each step of the path is related to a fault class, a detection mechanism, and a test; and to formalize such framework with probabilistic and efficiency models. The architecture that is to be considered is presented in Figure 1.

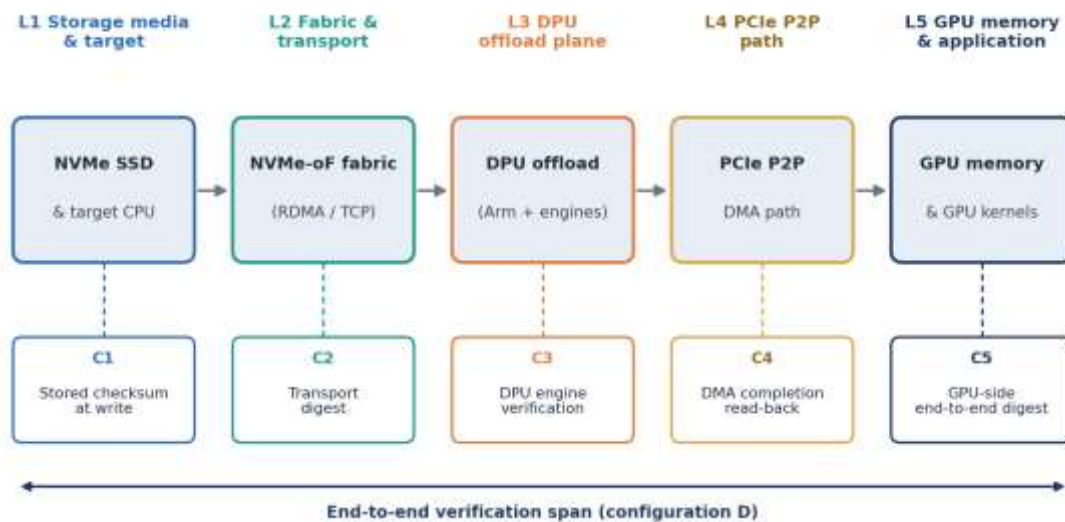


Figure 1. Layered Storage-to-GPU Path with Integrity Checkpoints C1 to C5

LITERATURE REVIEW

2.1 NVMe-oF and GPU-Direct Paths

Micro-benchmarks and a disaggregated key-value store [3] were used to characterize remote flash access for NVMe-oF over a 100 Gbit/s fabric compared to direct-attached NVMe and remote Internet Small Computer System Interface (iSCSI) storage. The protocol increased direct-attached latency by 11.7 μ s, while application throughput for the RocksDB-based store was found to be comparable to local access, and iSCSI's throughput was degraded by 40%. Previous research on iSCSI-based flash disaggregation had shown application throughput losses of around 20%. The numbers in Table 1 are used as a reference point when evaluating the integrity of any mechanism, as the primary value of the protocol is the minimal time needed to check. These figures are taken from Table 1 and represent a standard by which the integrity of any mechanism would be measured,

since the primary value of the protocol is the minimal time required to check. Fabric-level efficiency, however, doesn't indicate payload correctness, and digests and end-to-end verification take care of that.

Table 1. Remote Storage Access Compared with Direct-Attached NVMe

Access mode	Added latency vs. DAS	KV-store throughput vs. DAS	Earlier application- level loss	Fabric rate
Direct-attached NVMe (DAS)	Baseline	100%	—	—
NVMe-oF (RDMA)	+11.7 μ s	On par ($\approx$ 100%)	Negligible	100 Gbit/s
Remote iSCSI	Not tabulated	$\approx$ 60% (40% loss)	$\approx$ 20%	100 Gbit/s

Note. DAS = direct-attached storage; KV = key-value. Percentages are relative to DAS; the 60% value is derived from the reported 40% degradation. Source: synthesized from the reviewed performance characterization.

GDS eliminates CPU bounce buffer from data path by adding support for direct memory access between storage devices/ network adapters and GPU memory. The performance improvements of this arrangement over the traditional bounce-buffer path has been quantified in controlled experiments [4]. Since the direct path does not involve host memory, there is no host memory copy of the payload to inspect, checksum, or compare. This means that any verification must be moved to the DPU or to the GPU. Figure 2 shows that the reviewed work on the subject of integrity came after the work on the performance question had grown: the distribution of the literature over time shows that the first five of the 17 sources appeared in the same year, and that DPU offload and fleet-scale corruption analysis appeared together after DPU performance had been well established.

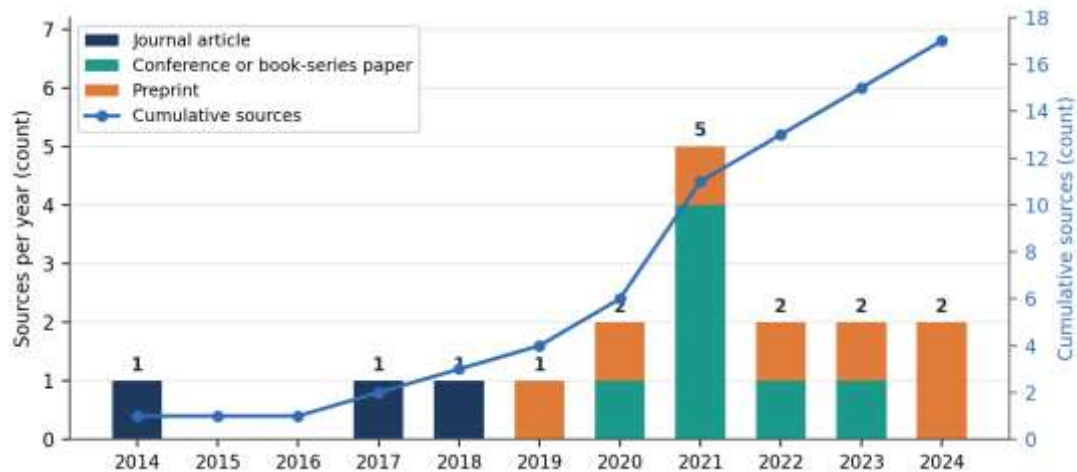


Figure 2. Distribution of the Reviewed Sources by Publication Year and Type

2.2 DPU Offload and Transport Behavior

The DPU offloads mainly come from high-performance computing. By offloading selected phases of distributed deep neural network training to the Arm cores on the BlueField-2 DPUs, as much as 15% of the overall training time was saved [5], showing the cores can handle data-handling

workloads that would traditionally be performed on the host. The implication of this result for integrity qualification is that offloaded phases run on data that has been staged to DPU memory and any corruption caused by a process running on the Arm-core or by a DMA engine is not visible to host-side integrity checks. Thus, offload adds to the collection of elements to be qualified, and the efficiency gain provided by offload must be reported in conjunction with the verification overhead which makes the efficiency gain meaningful.

Finally, the choice of transport further affects the qualification problem. NVMe-oF is supported by remote DMA (RDMA) fabrics (InfiniBand, RDMA over Converged Ethernet (ROCE), and iWARP), Fibre Channel, and Ethernet with the Transmission Control Protocol (TCP), which have varying throughput, CPU usage and configuration sensitivity over commodity networks [6]. A characterization methodology that takes CPU cores into consideration is therefore needed, because transports of the same bandwidth can use a very different number of cores. RDMA transports deliver link-level protection and completion semantics while TCP based transports support optional header and data digests. The qualification campaign should then be designed to log the digests that are enabled in each configuration tested, so that an unprotected configuration can be verified as protected by the digests that it enables. Table 2 summarizes the literature-to-concept mapping.

Table 2. Literature-to-Concept Mapping of the Reviewed Sources

Ref.	Evidence type	Layer and concept	Principal contribution
[2]	Preprint	L1 and L5, SDC testing	Contrast of out-of-production (Fleetscanner) and in-production (Ripple) detection over more than three years.
[7]	Preprint	L1, SDC at scale	Monitoring over 18+ months across hundreds of thousands of machines; hundreds of faulty processors found.
[3]	Journal article	L2, NVMe-oF	+11.7 μ s over DAS; throughput on par with local access; iSCSI degraded by 40% on a key-value store.
[4]	Conference paper	L4, GDS	Quantified performance gains of the direct storage-to-GPU path.
[5]	Conference paper	L3, DPU offload	Offload of training phases to BlueField-2; up to 15% improvement in training time.
[6]	Book-series paper	L2, transports	Effect of commodity networks on NVMe-oF disaggregation; core-aware characterization.
[1]	Symposium paper	L3, DPU architecture	Architecture of the DPU combining network interfaces, Arm cores, and offload engines.
[8]	Preprint (survey)	L3, SmartNIC taxonomy	Architectures, development models, and offload taxonomy (network, security, storage, compute).
[9]	Preprint	L3, eBPF	Safe and efficient remote code execution on disaggregated NVM storage.
[10]	Preprint	L4, GPU interconnects	Five interconnect types on six platforms; four non-uniform communication effects.

[11]	Journal article (tutorial)	L2, InfiniBand verbs	Integrated tutorial on InfiniBand, verbs, and MPI.
[12]	Conference paper	L3, offload benchmark	DPU-Bench micro-benchmark suite for SmartNIC offload efficiency.
[13]	Conference paper	L3, MPI offload	Nonblocking collectives on BlueField-2; overlap close to 100%.
[14]	Journal article	L5, GPU file system	GPUfs: file-system interface integrated with GPU programs.
[15]	Preprint (survey)	L4 and L5, GPU communication	Landscape of GPU-centric communication, host-controlled versus GPU-initiated.
[16]	Book-series paper	L2, NVMe/TCP	User-space approach to optimizing the NVMe-oF TCP transport.
[17]	Preprint	L2 and L3, pushdown	BPF-oF: up to 2.8 times throughput and 2.6 times tail latency across three storage systems.

Note. L1 to L5 denote the path layers of the qualification framework (Table 3). Source: synthesized from the reviewed studies.

METHODS

3.1 Review Protocol

A systematic review protocol was used for a set of sources (17): three journal articles, seven conference papers, seven preprints. The source had to include at least one of the following components of the storage-to-GPU path: remote storage protocols, GPU-direct data movement, DPU or SmartNIC offload, programmable storage functions, or silent corruption. Each source was categorized based on the offload taxonomy for network, security, storage, and compute functions [8] and on the path layer the evidence is relevant for, as shown in Table 2. Quantitative statements were extracted as numbers, and compared on a common baseline using experimental context. The findings from the qualitative data were categorized as enabling, constraining, or risk-bearing for integrity and discrepancies among sources were addressed by keeping the more conservative finding.

3.2 Layered Qualification Framework

The qualification framework divides the path into five layers (L1, L2, L3, L4 and L5), as is described in Table 3. A separate obligation of admission control exists for the DPU layer. Similar to what is done in extended Berkeley Packet Filter (eBPF) verification [9], execution of code provided by the application onto remote storage can be made safe and efficient, provided that the code is statically verified before it is allowed and then exercised with the code enabled or disabled on the same payloads in differential tests. If throughput does not increase, but there is a mismatch between the two outputs, the failure is considered a layer L3 integrity failure.

Table 3. Layer-Wise Qualification Matrix

Layer	Representative fault class	Detection mechanism	Qualification test
-------	----------------------------	---------------------	--------------------

L1 Storage media and target	Latent media faults; silent datapath corruption in processors	Stored checksums; pattern-sensitive scan tests	Offline pattern sweep (Fleetscanner-style) and in-production sampling (Ripple-style)
L2 Fabric and transport	Bit errors in flight; retransmission anomalies	Link-level CRC; transport digests; completion checks	Controlled error injection with digest verification and digest-enable audit
L3 DPU offload plane	Arm-core software faults; DMA-engine misconfiguration; unsafe offloaded functions	Engine-level CRC; verifier-based admission control	Offload-on versus offload-off differential test on identical payloads
L4 PCIe peer-to-peer path	Misrouted peer-to-peer DMA; topology-dependent behavior	Topology audit; completion read-back	Test matrix over every GPU–NIC pair
L5 GPU memory and application	GPU datapath SDC; stale or partial reads	Device-side digest kernels; redundancy	Recomputation of digests in GPU memory against the stored value

Note. CRC = cyclic redundancy check; SDC = silent data corruption. Source: synthesized from the reviewed studies.

The evaluation of five types of GPUs and interconnects on six platforms resulted in four different effects of non-uniform communication, three of which were induced by NVLink topology and one by PCIe chipset design [10] and made it necessary to qualify the Layer 4 with a topology-aware test matrix. Each GPU/NIC combination is then considered a test point. To combine the layers quantitatively, p_i is given as the probability of corruption at layer i per transferred block, and c_i as the coverage of the checks that span that layer. If the corruption is not detected, the residual probability is as per Equation (1); if it is detected, the probability of any block being corrupted is p/n ; and the expected number of undetected events for a data volume V and block size s is as per Equation (3). Four protection configurations of Equation (3) are plotted in Figure 3.

$$P_{\text{und}} = 1 - \prod_{i=1}^5 [1 - p_i(1 - c_i)] \quad (1)$$

$$c = 1 - 2^{-n} \quad (2)$$

$$E[U] = \left(\frac{V}{s}\right) \cdot P_{\text{und}} \quad (3)$$

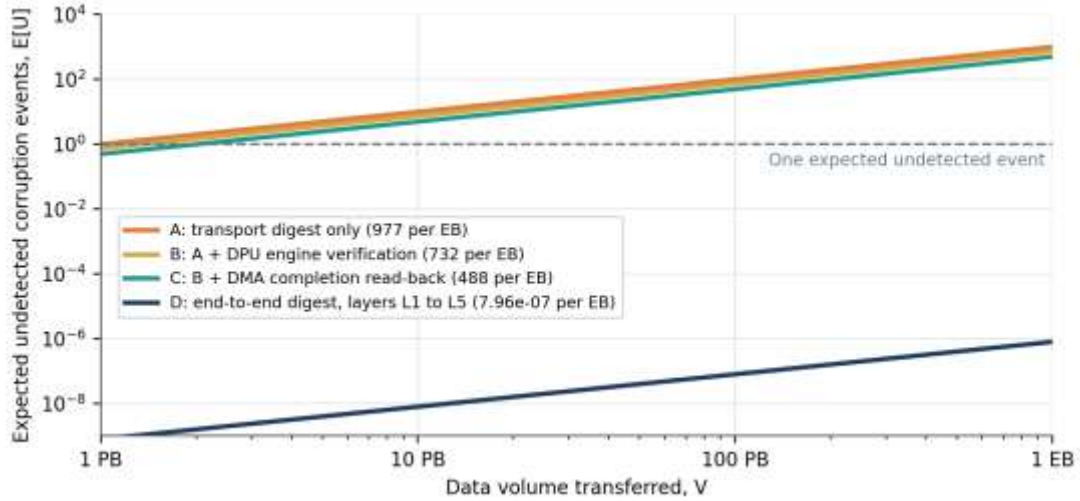


Figure 3. Expected Undetected Corruption Events under Four Protection Configurations (Model-Derived)

The verbs level of qualification takes advantage of a queue-pair and completion model of InfiniBand that indicates successful completion of a work request, but not necessarily that the delivered bytes are valid [11]. A test harness thus combines each completion with its own digest comparison, and keeps track of the overhead ratio of Equation (4) that is incurred in the checks themselves, and the other terms of path latency: the media, fabric, DPU and DMA. The efficiency of offloading is measured by the overlap measure (5), where T_{ovl} is the total time, including offloading, T_{comp} the computing time, and T_{pure} the pure communication time. A campaign needs to be extended until the target confidence α is achieved, and Equation (6) gives the number of passes which is required to be completed for a per-pass detection probability d and target confidence α , which is plotted in Figure 4 (59 passes for $d = 0.05$ and $\alpha = 0.95$).

$$\eta = \frac{T_{verify}}{T_{media} + T_{fabric} + T_{dpu} + T_{dma} + T_{verify}} \quad (4)$$

$$O = \max\left(0, 1 - \frac{T_{ovl} - T_{comp}}{T_{pure}}\right) \times 100 \quad (5)$$

$$k \geq \left\lceil \frac{\ln(1-\alpha)}{\ln(1-d)} \right\rceil \quad (6)$$

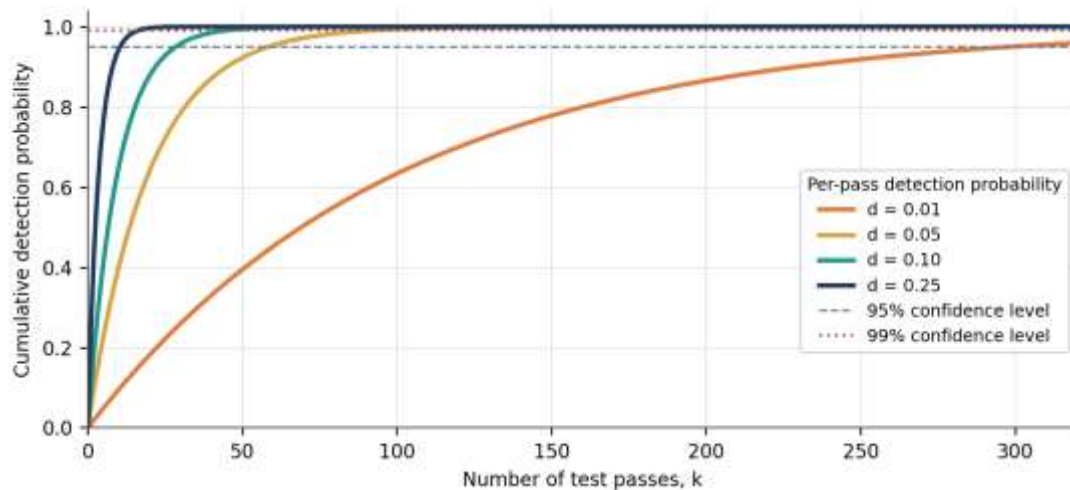


Figure 4. Cumulative Detection Probability as a Function of Test Passes (Model-Derived)

RESULTS AND DISCUSSION

4.1 Offload Efficiency Evidence

DPU-Bench developed a micro-benchmark technique for measuring the efficiency of offloading progress on the SmartNICs, which allows the separation of the benefits that can be achieved from offloaded progress from the cost of the offload itself [12]. The same separation is used here for the synthesis of reported results, in two ways: first, by normalizing with the speedup S of Equation (7), and second, by normalizing with the relative improvement Δ of Equation (8) with T_{base} and T_{opt} the baseline and optimized execution times. None of the reviewed studies report the verification overhead of Equation (4), and as compiled in Table 4, these studies report improvement ranging from 15% in reducing training time to a factor of 2.8 in reducing key-value throughput. The main gap in evidence is this omission: there is extensive evidence measuring offload efficiency, with no evidence measuring integrity efficiency.

$$S = \frac{T_{base}}{T_{opt}} \quad (7)$$

$$\Delta = \frac{T_{base} - T_{opt}}{T_{base}} \times 100 \quad (8)$$

Table 4. Offload Efficiency Evidence and Integrity Implications

Study focus	Offloaded or measured function	Metric	Reported effect	Integrity implication
DNN training on BlueField-2	Selected training phases on Arm cores	Training time	Up to 15% improvement	Staged data in DPU memory requires verification
Nonblocking MPI collectives	Broadcast and allgather progression	Computation-communication overlap	Close to 100% for large messages	Chunked buffers reside outside host visibility
BPF-oF pushdown	eBPF storage functions at the remote target	Throughput ; tail latency	Up to 2.8 times; up to 2.6 times	Pushed functions need admission verification

NVMe-oF baseline	Remote access protocol	Latency; KV throughput	+11.7 μ s; on par with DAS	Digest use must be recorded per run
Fleet SDC testing	Out-of-production and in-production tests	Monitoring scope	18+ months; hundreds of thousands of machines	Compute silicon may corrupt silently

Note. Effects are as reported by the respective sources. Source: synthesized from the reviewed studies.

For large messages, offloading the nonblocking versions of the Message Passing Interface (MPI) collectives (MPI_Ibcast and MPI_Iallgather) into the BlueField-2 DPU adapter with 8 Arm cores led to near 100% communication to computation (C2C) overlap [13]. This is accomplished by the use of message chunking and staging in DPU memory, and is also where payload bytes are not visible to the host. On a common relative scale, as seen in Figure 5, the 15% training time reduction, 61.5% reduction in tail latency, 66.7% throughput advantage of NVMe-oF over iSCSI, and 180% throughput gain from storage-function pushdown are achieved. Such gains are worth offloading but also constitute the price limit for verification to remain within.

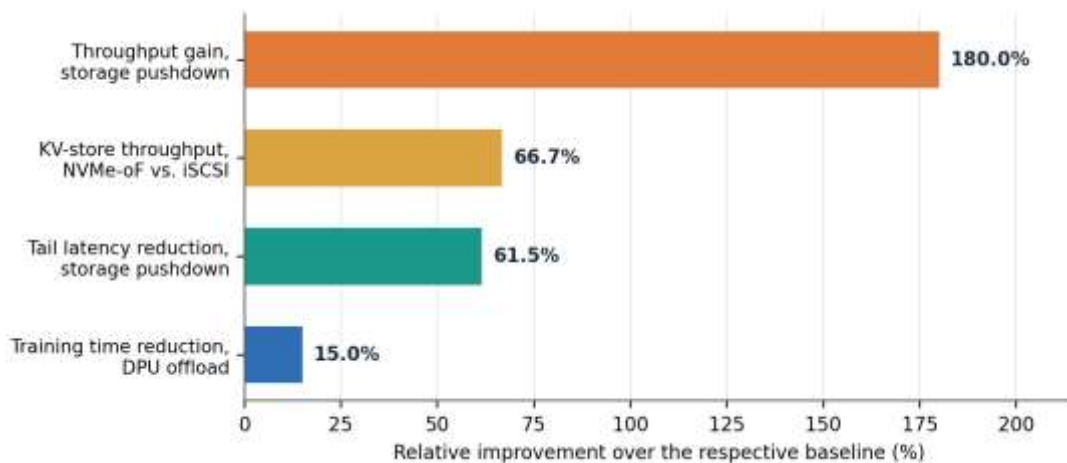


Figure 5. Relative Improvements Reported across the Reviewed Offload and Protocol Studies

4.2 Quantitative Integrity Model

The path is ultimately passed to the GPU, and the inclusion of file-system semantics in GPU programs, such as GPUfs, demonstrates that GPU code can directly make file operations [14]. In this way, it is possible to realize a capability that is practical on the device: a digest kernel could be implemented in the GPU and recalculate a checksum on received data and compare it with the one computed at the time of writing. The impact of such end-to-end coverage was assessed using Equations (1) to (3) with illustrative corruption probabilities of 10–12 per 4 KiB block for each layer (except the fabric, which has a probability of 10–11) and 32-bit digests. As shown in Table 5, the achieved residual probability is around 4.0×10^{-12} per block for transport-only protection, and around 8.0×10^{-7} per block for end-to-end protection (a reduction by a factor of about 1.2×10^9).

Table 5. Model-Derived Residual Risk by Protection Configuration

Configuration	Layers spanned by digests	Digest checks	Residual P_{und} per block	Expected events per EB	Reduction vs. A
---------------	---------------------------	---------------	-------------------------------------	------------------------	-----------------

A: transport digest only	L2	1	4.0×10^{-12}	977	$1.0 \times$
B: A + DPU engine verification	L2, L3	2	3.0×10^{-12}	732	$1.3 \times$
C: B + DMA completion read-back	L2 to L4	3	2.0×10^{-12}	488	$2.0 \times$
D: end-to-end digest	L1 to L5	5	3.3×10^{-21}	8.0×10^{-7}	$1.2 \times 10^9 \times$

Note. Computed from Equations (1) to (3) with 4 KiB blocks, 32-bit digests, and illustrative corruption probabilities (10^{-12} per layer; 10^{-11} for the fabric). The values are model outputs, not measurements. EB = exabyte.

There are two main types of GPU-centric communication: designs using the host to control data movement, and designs using the GPU threads to initiate data movement. The difference is used to place the verification point and drives four deployable configurations to be scored based on five criteria: integrity coverage, latency overhead, host-CPU relief, topology robustness and evidence maturity. The aggregate score is the weighted sum of Equation (9), with weights of 0.30, 0.20, 0.15, 0.15, and 0.20, respectively. The evidence scores obtained from the reviewed evidence are displayed as ordinal scores in Figure 6. The top score (3.80) achieved was from the DPU-offloaded GDS configuration, followed closely by DPU offload with storage pushdown (3.50), host RDMA with GDS (2.95), and host-bounce NVMe/TCP (2.85). The ranking is not measurement and is sensitive to the weights.

$$Q = \sum_{j=1}^5 w_j s_j \quad (9)$$

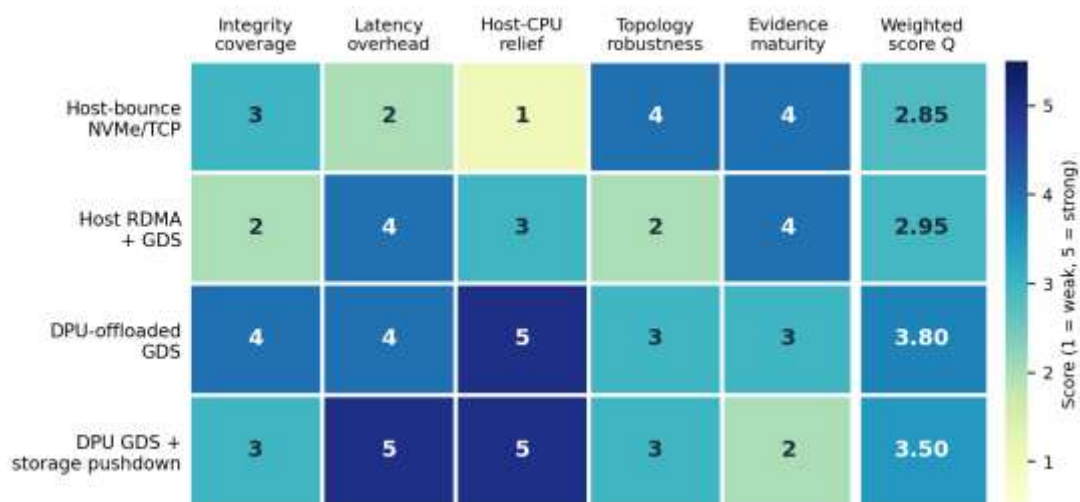


Figure 6. Ordinal Scoring of Four Deployable Configurations against Five Criteria

4.3 Sequenced Campaign and Open Challenges

Proposal of an NVMe/TCP transport for optimizing NVMe-oF over TCP [16] highlights the fact that the software path, and not the wire protocol, is often the factor that determines cost and observability. An advantage of a user-space transport is that digests can be calculated and recorded at a known location in the stack and failures can be matched with detailed commands. The overall evidence thus makes sense in a sequential campaign: transport digests are validated first (L2), offload-plane differential tests follow (L3), topology-aware peer-to-peer tests are performed next (L4), and finally, end-to-end validation is executed via the GPU-side operation (L5). Whereas each stage provides the localization information required to understand the failure encountered at the

next stage, thereby reducing the debugging time that is known to be necessary for a failure due to silent corruption.

For RocksDB, by integrating with three storage systems and implementing storage-function pushdown for NVMe-oF, the performance was boosted by up to 2.8 times in terms of throughput and up to 2.6 times in terms of tail latency [17]. The same report pointed to disk arrays, redundant array of independent disks (RAID) schemes and encrypted storage as capabilities that were not yet resolved, and execution on a SmartNIC as a promising way forward. Each of these limitations has direct qualification implications as outlined in Table 6: pushed-down functions must be verified prior to admission, encryption and redundancy layers introduce additional transformation stages that require additional verification, and heterogeneous SmartNIC toolchains can result in repeatability challenges. In future work reviewed in the presented evidence there are standardized integrity benchmarks for offload engines, systematic screening for silent faults of DPU Arm cores, and measuring the verification overhead in addition to offload gains.

Table 6. Open Challenges and Qualification Responses

Challenge	Evidence in the reviewed studies	Qualification response
Missing integrity metrics in offload studies	Gains of 15% and 2.8 times reported without verification overhead (Table 4)	Report η of Equation (4) with every offload result
Silent faults in compute silicon	Hundreds of faulty processors found over 18+ months	Screen DPU Arm cores and GPUs with out-of-production and in-production tests
Topology-dependent behavior	Four non-uniform effects across six platforms	Treat each GPU-NIC pair as a separate test point
Heterogeneous SmartNIC toolchains	Differing architectures and development models in the surveyed literature	Fix and record SDK versions in the test harness
Unsupported pushdown features	RAID and encrypted storage unresolved in BPF-oF	Treat as transformation stages with checks before and after
Host-invisible payloads under GDS	Bounce buffer removed from the data path	Relocate digest verification to the DPU or the GPU

Note. Source: synthesized from the reviewed studies.

CONCLUSION

The review highlights that characteristics of performance, transport completion, or lack thereof do not constitute a proof of the integrity of the GPU-direct storage paths with NVMe-oF and DPU offload. Each transition of data from a storage media to the GPU memory is another point where corruption can occur, and offload moves payloads away from memory locations the host-side checks. The proposed five-layer framework is accompanied by a probabilistic model: Only verification over the full path reduces residual risk to a negligible level, associating each stage with a fault class, a detection mechanism, and a qualification test. The lack of reported overhead associated with verification is cited as the main evidence gap in the reviewed offload studies.

Offload studies are suggested to include the overhead for verification along with throughput and latency improvements, DPU Arm cores to be integrated into fleet screening programs and fault-

injection campaigns to replace the illustrative per-layer probabilities that are used here with measured probabilities.

ACKNOWLEDGEMENTS

Sponsor and financial support acknowledgments are to be completed by the authors prior to submission.

REFERENCES

- [1] I. Burstein, "NVIDIA data center processing unit (DPU) architecture," in *2021 IEEE Hot Chips 33 Symposium (HCS)*, IEEE, 2021, pp. 1–20. doi: 10.1109/HCS52781.2021.9567066.
- [2] H. D. Dixit, L. Boyle, G. Vunnam, S. Pendharkar, M. Beadon, and S. Sankar, "Detecting silent data corruptions in the wild," 2022. doi: 10.48550/arXiv.2203.08989.
- [3] Z. Guz, H. Li, A. Shayesteh, and V. Balakrishnan, "Performance characterization of NVMe-over-Fabrics storage disaggregation," *ACM Trans. Storage*, vol. 14, no. 4, Art. no. 31, 2018, doi: 10.1145/3239563.
- [4] D. Inupakutika, B. Davis, Q. Yang, D. Kim, and D. Akopian, "Quantifying performance gains of GPUDirect Storage," in *2022 IEEE International Conference on Networking, Architecture and Storage (NAS)*, IEEE, 2022. doi: 10.1109/NAS55553.2022.9925516.
- [5] A. Jain, N. Alnaasan, A. Shafi, H. Subramoni, and D. 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)*, IEEE, 2021, pp. 17–24. doi: 10.1109/HOTI52880.2021.00017.
- [6] A. Kashyap, S. Guhani, and X. Lu, "Impact of commodity networks on storage disaggregation with NVMe-oF," in *Benchmarking, Measuring, and Optimizing*, F. Wolf and W. Gao, Eds., in *Lecture Notes in Computer Science*, vol. 12614. Springer, 2021, pp. 41–56. doi: 10.1007/978-3-030-71058-3_3.
- [7] H. D. Dixit *et al.*, "Silent data corruptions at scale," 2021. doi: 10.48550/arXiv.2102.11245.
- [8] E. Kfoury, S. Choueiri, A. Mazloum, A. AlSabeh, J. Gomez, and J. Crichigno, "A comprehensive survey on SmartNICs: Architectures, development models, applications, and research directions," 2024. doi: 10.48550/arXiv.2405.09499.
- [9] K. Kourtis, A. Trivedi, and N. Ioannou, "Safe and efficient remote application code execution on disaggregated NVM storage with eBPF," 2020. doi: 10.48550/arXiv.2002.11528.
- [10] A. Li *et al.*, "Evaluating modern GPU interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect," 2019. doi: 10.48550/arXiv.1903.04611.
- [11] P. MacArthur, Q. Liu, R. D. Russell, F. Mizero, M. Veeraraghavan, and J. M. Dennis, "An integrated tutorial on InfiniBand, verbs, and MPI," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2894–2926, 2017, doi: 10.1109/COMST.2017.2746083.
- [12] B. Michalowicz, K. K. Suresh, H. Subramoni, D. K. Panda, and S. Poole, "DPU-Bench: A micro-benchmark suite to measure offload efficiency of SmartNICs," in *Practice and Experience in Advanced Research Computing (PEARC '23)*, ACM, 2023, pp. 94–101. doi: 10.1145/3569951.3593595.
- [13] N. Sarkauskas, M. Bayatpour, T. Tran, B. Ramesh, H. Subramoni, and D. K. Panda, "Large-message nonblocking MPI_allgather and MPI_ibcast offload via BlueField-2 DPU," in *2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, IEEE, 2021, pp. 388–393. doi: 10.1109/HiPC53243.2021.00054.
- [14] M. Silberstein, B. Ford, I. Keidar, and E. Witchel, "GPUs: Integrating a file system with GPUs," *ACM Trans. Comput. Syst.*, vol. 32, no. 1, Art. no. 1, 2014, doi: 10.1145/2553081.
- [15] D. Unat *et al.*, "The landscape of GPU-centric communication," 2024. doi: 10.48550/arXiv.2409.09874.
- [16] Z. Yang, Q. Wan, G. Cao, and K. Latecki, "uNVMe-TCP: A user space approach to optimizing NVMe over fabrics TCP transport," in *Internet of Vehicles. Technologies and Services Toward Smart Cities (IOV 2019)*, C.-H. Hsu, S. Kallel, K.-C. Lan, and Z. Zheng, Eds., in *Lecture Notes in Computer Science*, vol. 11894. Springer, 2020, pp. 125–142. doi: 10.1007/978-3-030-38651-1_13.
- [17] I. Zarkadas *et al.*, "BPF-oF: Storage function pushdown over the network," 2023. doi: 10.48550/arXiv.2312.06808.