

Cross-Protocol Data-Integrity Validation During Live Migration Between NVMe and Fibre Channel Storage Paths

Gagan Vyas

Member of technical staff, Pure storag, Santa clara California, United States

Article Info	ABSTRACT
Article history: Received December, 2025 Revised December, 2025 Accepted December, 2025 Keywords: Checksum Validation; Data Integrity; Fibre Channel; Live Migration; NVMe Over Fabrics; Pre-Copy Migration; Silent Data Corruption; Storage Virtualization; Systematic Review	Any data corruption that happens between the two storage paths while they are both active can go unnoticed. The 20 records supplied were used to conduct a systematic review of how data-integrity validation can be organized as the volumes move between Fibre Channel (FC) and Non-Volatile Memory Express (NVMe) paths. Evidence of the effects of storage errors, flash media, fabric performance, protocol digests, and migration mechanics was aggregated and treatment was added to an analytical model of validation cost and residual risk. Field data revealed 0.86% disks with checksum mismatches on nearline disks and 0.065% FC enterprise disks with checksum mismatches; 3.45% disks with latent sector errors. The application throughput was 20% lower with remote flash access over the Internet Small Computer System Interface (iSCSI) than without it, while NVMe over Fabrics introduced 11.7 μs of latency. With a corruption probability of 10^{-7} per block, the model predicted that about 6.25×10^{-9} blocks would be corrupted each time 1 TiB was migrated, but, as the 4 GiB/s full inline validation showed, pre-copy migration was not convergent. A compromise was reached by sampling the digest on the fly during the copying process and then comparing the final digest with the on disk digest. The main research gap identified is having no direct empirical study on FC-to-NVMe migration. <i>This is an open access article under the CC BY-SA license.</i> 

Corresponding Author: Name: Gagan Vyas Institution: Member of technical staff, Pure storag, Santa clara California, United States Email: gaganvyas@gmail.com
--

1. INTRODUCTION

Silent data corruption is a problem that always affects stored data without an obvious error message being returned to the applications. More than 400,000 checksum mismatches have been observed from a field study of 1.53 million disk drives in production storage systems over 41 months of observation [1]. Nearline disks had an order of magnitude higher number of mismatches, 0.86% of about 358,000 disks compared to

0.065% of about 1.17 million enterprise Fibre Channel (FC) disks. The events also tended to be grouped together in time and space and were not independent events on different disks. Integrity is not something that one can set only at the time of writing. Any operation which reads, moves and rewrites large amounts of storage uses an entire storage stack, and thus has the chance to uncover the damage and spread it.

Live - Moves a running workload between hosts or storage back ends without

stopping the service. The ability enhances manageability and fault-tolerance, but it comes at a price. Experiments with a multi-tier web application on Xen virtual machines (VMs) demonstrated up to a 3s of service downtime in a nearly oversubscribed system serving 600 users simultaneously, and violation of SLA when two migrations were carried out back-to-back [2]. Adding data-integrity checking adds work to an already timing-sensitive process. The validation procedures then need to be based on the same copy bandwidth, downtime and dirty data convergence budget associated with the migration, and not as an additional audit phase added on.

Many of the current data centers introduce and mix legacy block paths with Non-Volatile Memory Express (NVMe) over Fabrics (NVMe-oF), which is a protocol to transfer and execute NVMe commands over multiple transports such as remote direct memory access (RDMA) fabrics, FC, and Ethernet with TCP [3]. When a volume is transferred from an FC-attached Small Computer System Interface (SCSI) path to an NVMe-oF path it switches between a change of command set, queueing model, and error reporting model. The literature on corruption, fabric performance and migration mechanics are treated separately. In the present review, these streams are combined, an analytical

validation framework is developed, and a trade-off relation between the detection strength and the convergence of the migrations is quantified. The work is guided by 3 questions: Where can the integrity be proven, what is the cost of verification and what is the residual risk?

2. LITERATURE REVIEW

2.1 Latent Errors and Silent Corruption

Latent sector errors (LSEs) are errors that go unnoticed until the affected sectors are read. A study of the production systems for 1.53 million disks over 32 months revealed that 3.45% of the disks experienced at least one LSE, that enterprise class disks were not as likely to suffer from LSEs as nearline disks, and that the fraction of disks with affected disks increased with disk capacity [4]. Disk with LSEs had less than 50 errors in more than 80% of them and an already failed disk was more likely to fail again. These prevalence rates are summarized in Figure 1 along with the checksum mismatch rates mentioned above. A complete read pass during migration also serves as a discovery event, since any errors that are found during read will have to be addressed later during access.

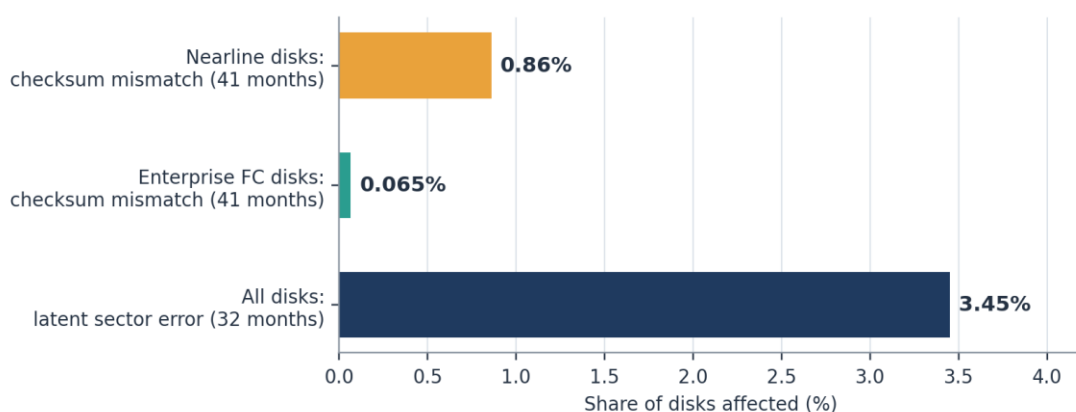


Figure 1. Prevalence of Checksum Mismatches and Latent Sector Errors in Field Data

The data collected from LSE were analysed further to look at the patterns of errors and how protection mechanisms should respond. The

locality of errors led to the development of protective policies, including proactive scrubbing which reads disk data in the background to detect LSEs

before a second fault caused data loss [5]. The three field studies are summarized in Table 1. Most of the time it is agreed that what matters is not the amount of errors, but the time of detection. If each block

read is compared with a reference that is independent of the other, a controlled migration that reads every block once acts as an accelerated scrub.

Table 1. Field Evidence on Latent Errors and Silent Corruption

Study	Population and window	Principal findings	Implication for migration
[1]	1.53 million disks; 41 months	>400,000 checksum mismatches; 0.86% of about 358,000 nearline disks versus 0.065% of about 1.17 million enterprise FC disks	Rare per disk but certain at fleet scale; clustering invalidates independence assumptions
[4]	1.53 million disks; 32 months	3.45% of disks developed LSEs; over 80% of affected disks had fewer than 50 errors; prevalence rose with capacity	A full-volume read acts as a discovery event for latent faults
[5]	Field LSE data from a large drive population	Statistical characterization of LSE locality; protective policies including scrubbing	Detection latency, not raw frequency, determines exposure

Source: Synthesized from the cited studies. LSE = latent sector error.

File systems also affect the visibility of storage faults. A failure-policy analysis of commodity file systems revealed that there exists a fail-partial model, where individual blocks fail or may return corrupted data, but the remaining blocks keep functioning, and that the detection and recovery policies were often not consistent or were missing [6]. This is a model that fits migration scenarios directly: A copy engine will face partial failures when reading and writing bulk data. Corrupted blocks can be copied to the destination without any changes, and thus will be indistinguishable from legitimate data. A migration framework, then, should include explicit failure detection logic and a specific policy for every failure class, and not rely on the implicit behavior of the guest file system.

2.2 Flash Media and Disaggregated NVMe Paths

The reliability of media on the destination side is also uncertain. A field data analysis of the failure rates of a large fleet of flash-based solid-state drives (SSDs) for a period of nearly four years and many millions of operational hours

has revealed that failure rates do not increase monotonically over time, but rather exhibit distinct phases, which correspond to the emergence and detection of failures [7]. Failure rates were also affected by temperature, data format and controller operation. A newly created NVMe volume can thus not be assumed to be more reliable than an old legacy volume and the early-life period is likely relevant since migration targets are often freshly provisioned devices, and they tend to be inundated with writes shortly after creation.

The performance benefit of NVMe is significant enough to impact migration. NVMe-backed database applications had up to eight times the client-side performance of enterprise class SATA-based SSDs [8] in an in-depth analysis of the NVMe drives, including instrumentation of access times through the system software stack. This quick change of bottleneck to the source path, network, or validation engine makes this a destination that can absorb writes. Thus, the hashing and comparing speed of the blocks becomes a first-order design factor, and an adequate validation engine

for an FC source can throttle an NVMe destination.

This was done before the general adoption of NVMe-oF. Experiments using commodity networks for remote flash access over the Internet Small Computer System Interface (iSCSI) revealed 20% reduction in the application level throughput, but this can be overcome by scaling out the flash and processor resources [9]. The reason for the overhead was mostly due to protocol processing and not the network. The results set the baseline to which fabric-native NVMe access is measured, and reveal that any migration of NVMe access to another protocol layer through a SCSI-class intermediary incurs protocol-processing costs that cannot be avoided if NVMe access is native.

2.3 Live Migration Techniques

The frequency of copying is dictated by memory and state transfer strategies and thus corresponds to the frequency of data validation. Pre-copy migration involves copying the memory multiple times during the VM's running time, followed by a short stop-and-copy period. Post copy migration: memory transfer is delayed until the target processor state is reached. A Xen

implementation on a Gigabit LAN demonstrated that post-copy resulted in better pages transferred, total migration time, and network overhead, compared to pre-copy, for various workloads [10]. Under post-copy, each page is transferred at most once – this also makes it easier to keep track of the integrity, but it means the network can produce page faults during an execution, leaving the source open to failure.

A comprehensive survey of migration challenges was done by giving an overview of the techniques from memory, storage and network perspective and documenting open problems [11]. Table 2 shows the major strategies and outlines their integrity implications. Migration to storage, especially, is seldom addressed by the memory-centric algorithms: block devices need to be copied, written history tracked, and operated in the same order with the same consistency constraints. This synchronization also includes translating the semantics of the commands from the source protocol to the destination protocol (protocol mapping), in which cross-protocol validation needs to be anchored.

Table 2. Live Migration Strategies and Their Integrity Implications

Strategy	Mechanism	Reported evidence	Integrity implication
Pre-copy	Iterative copy of changed data while the VM runs, then a short stop-and-copy phase	Downtime up to 3 s with 600 concurrent users [2]	Retransmitted blocks require repeated validation
Post-copy	Processor state is sent first and memory is fetched afterward	Improved pages transferred, total time, and network overhead on Xen [10]	Each page moves once, but network faults expose the running VM
Hybrid	Limited pre-copy followed by post-copy	Reviewed in Section 4.4	Bounds the number of rounds that need validation

Source: Synthesized from the cited studies and the survey in [11].

3. METHODS

3.1 Review Protocol

The review was conducted according to corpus-defined procedure. The 20 provided records were pre-filtered by four criteria: relevance to storage integrity, protocol behavior,

migration mechanics, or validation overhead; quantitative or mechanistic evidence; empirical or survey character; applicability to block storage. The 20 records were all found to meet the criteria, and no record was rejected, as shown in Figure 2. Cost-oriented fields

(e.g., duration in the symbol, downtime, resource consumption, etc.) of a survey of live migration costs [12] were used for data extraction, together with integrity-

oriented fields (e.g., error class, detection layer, etc.). Table 3 illustrates the links between each record and the concepts used in the synthesis.

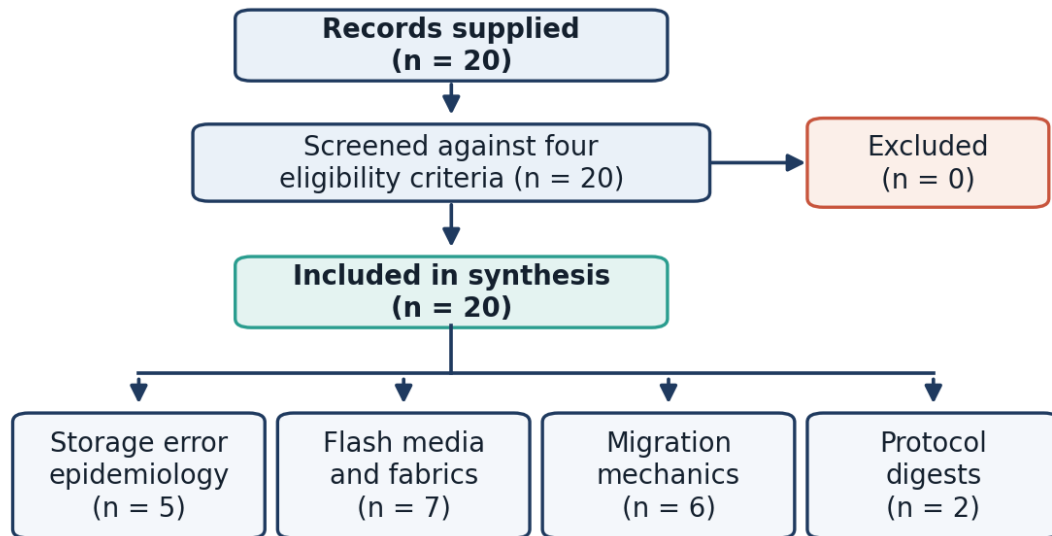


Figure 2. Review Flow and Evidence Streams

Table 3. Mapping of Reviewed Literature to Analytical Concepts

Concept	Records	Role in the framework
Silent corruption and latent errors	[1], [4], [5], [6]	Defines threat classes and detection latency
Flash media reliability	[7], [13]	Source of destination-side residual errors
NVMe, disaggregation, and fabrics	[3], [8], [9], [14], [15]	Path performance and transport options
System-level performance faults	[16]	Interaction of fail-slow behavior with validation
Protocol digests	[17], [18]	Digest strength and layer coverage
Migration mechanics and cost	[2], [10], [11], [12], [19], [20]	Convergence, downtime, and strategy choice

Source: Classification performed during the review; all 20 records are mapped.

3.2 Analytical Framework

The validation framework is based on block level digests. A digest function H returns a mismatch indicator for the source content s_b and the destination content d_b of a block b , and the sum of the digest functions for N blocks returns the number of mismatches M (Equation 1). The 32-bit cyclic redundancy check (CRC) used for iSCSI in this manner, and other 32-bit checksums, have a probability of 2^{-32} that a random corruption will go undetected [17]. The expected number of

undetected corrupt blocks is then given by Equation 2, where V is the size of the volume, S_b the size of the blocks, p_c the probability of corruption per block, and k the width of the digest in bits. The probability that a fraction f of blocks is sampled and all m of them are corrupted is given by Equation 3. Figure 3 shows the three validation points in the dual-path architecture.

$$M = \sum_{b=1}^N \mathbb{I}(H(s_b) \neq H(d_b)) \quad (1)$$

$$E_{ud} = \frac{V}{S_b} p_c 2^{-k} \quad (2)$$

$$P_{miss} = (1 - f)^m \quad (3)$$

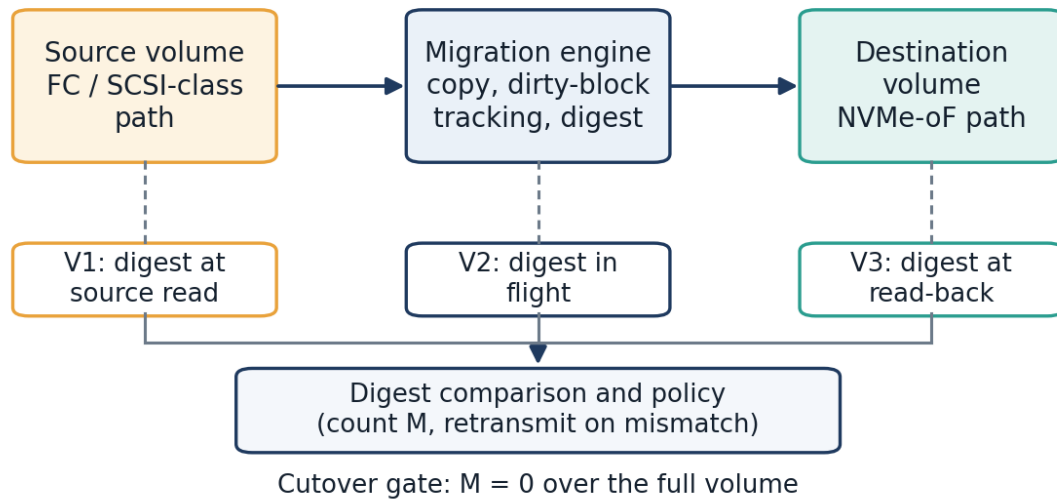


Figure 3. Dual-Path Migration Architecture With Three Validation Points

4. RESULTS AND DISCUSSION

4.1 Residual Corruption Risk

Flash error characterization reveals the classes of errors that can cause the occurrence of corruption candidates at a destination: program and erase cycling errors, program disturbance, cell-to-cell interference, data retention loss, and read disturbance, which are all handled by error correction and controller policies that are unknown to a migration engine [13]. The amount

represented by pc is therefore the only amount that the engine will see. Fig. 4 shows the evaluation of Equation 2 for volumes ranging from 1 to 1,000 terabytes (TiB) and per-block corruption probability ranging from 10^{-9} to 10^{-6} with 4 kilobyte (KiB) blocks. The expected number of undetected blocks is 6.25×10^{-5} , which means that the risk of random error with CRC32C is extremely low when compared to corruption occurring at the source, which a digest comparison can reproduce, not detect.

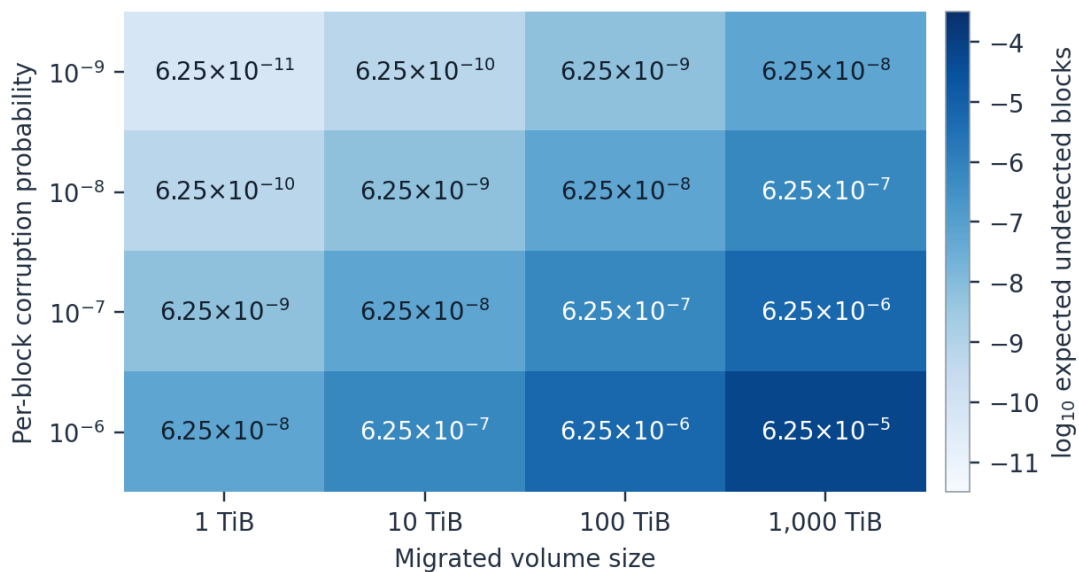


Figure 4. Expected Undetected Corrupt Blocks Under 32-Bit Digest Validation

There are no unbounded increases in the strength of the detection because there is an interaction between the validation load and the behavior of the hardware. The study of 114 reports of fail-slow hardware incidents among 14 institutions revealed that hardware was still working at much lower than expected speeds, and that it took hours or even months to diagnose the incident. The study of 114 reports covering 14 institutions found that hardware continued to operate at much

lower than expected speeds and that hours or even months were needed to identify the incident. Slow validation reads on such components can be confused for protocol errors. Table 4 illustrates that sampling only a tenth of blocks will miss 50 blocks with a 0.5% probability and miss a group of 5 blocks with a 59% probability, which means that sampling is not enough for clustered corruption and a last comparison of the whole volume should be done.

Table 4. Probability That Sampled Validation Misses All Corrupt Blocks (%)

Validated fraction f	$m = 1$	$m = 5$	$m = 20$	$m = 50$
0.5	50.0	3.1	<0.01	<0.01
0.25	75.0	23.7	0.32	<0.01
0.1	90.0	59.0	12.2	0.52
0.05	95.0	77.4	35.8	7.7

Note: Values are computed from Equation (3) under independent block sampling; m denotes the number of corrupt blocks.

4.2 Protocol-Path Overhead

The overhead gap typical of SCSI-class-remote storage is closing in the case of NVMe access via fabric. NVMe-oF over an RDMA fabric was characterized, and it was found that the access latency of remote access to NVMe-oF was 11.7 μ s, which is very close to the access latency of local NVMe-oF devices [14]. This evidence, combined with the above-mentioned iSCSI and media results, is summarized in Table 5. While it is a very small latency compared to the 1 millisecond or less access time for rotating media, it is a measureable part of the access time for fast NVMe access devices, meaning that each software hop added, such as computing a digest, shows up on an NVMe path.

The characterization started with micro-benchmarks and extended to a

disaggregated key-value store on a system capable of 100 Gbit/s, and demonstrated no degradation to the application for NVMe-oF versus the 20% degradation previously noted for iSCSI [15]. Better storage server scalability was also possible because of the reduced protocol-processing. The cost of the consequence for validation is given by the expression in equation 4, which is a function of the raw path bandwidth B , the hashing rate R_h , and the fraction of paths validated f . As illustrated in figure 5, assuming a bandwidth of 3.0 gibibytes per second (GiB/s), full validation will reduce the effective copy rate to 1.71 GiB/s, while validation of just 10% of the blocks will keep the rate at 2.79 GiB/s.

$$B_{\text{eff}} = \frac{1}{\frac{1}{B} + \frac{f}{R_h}} \quad (4)$$

Table 5. Protocol-Path Overhead Evidence and Model-Derived Validation Cost

Path or configuration	Metric	Reported or modeled value	Basis
NVMe versus enterprise SATA SSD	Client-side database performance	Up to 8× higher	[8]
Remote flash over iSCSI	Application-level throughput	20% lower	[9]

Path or configuration	Metric	Reported or modeled value	Basis
NVMe-oF versus direct-attached NVMe	Added access latency	11.7 μ s	[14]
NVMe-oF versus iSCSI, key-value store	Application-level degradation	Negligible for NVMe-oF	[15]
Full validation at $R_h = 4$ GiB/s	Effective copy rate	1.71 GiB/s (from 3.0)	Equation (4)
10% sampled validation at $R_h = 4$ GiB/s	Effective copy rate	2.79 GiB/s (from 3.0)	Equation (4)

Source: Reported values from [8], [9], [14], [15]; the last two rows are model outputs with an assumed bandwidth of 3.0 GiB/s.

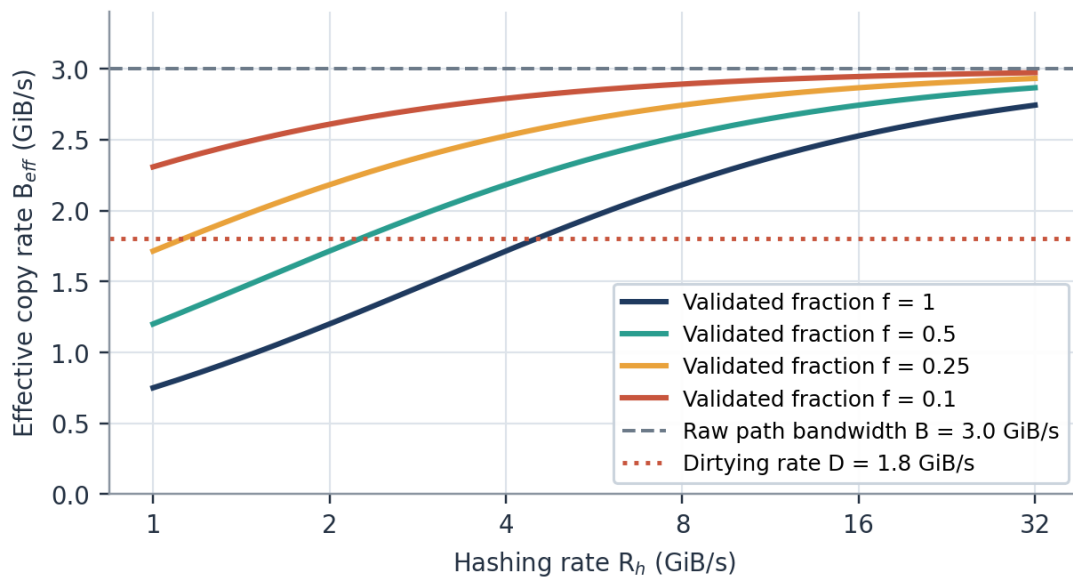


Figure 5. Effective Copy Rate as a Function of Hashing Rate and Validated Fraction

4.3 Digest Coverage Across Protocol Paths

The protection of the transport segment, but not the complete path, is provided by protocol level digests. The consolidated iSCSI specification specifies the optional header and data digests, which are negotiated per session and calculated using CRC32C, to allow for the detection of data corruption over the lower-layer that is not detected by the lower-layer checks [18]. Table 6 shows

the validation layers for the legacy and NVMe-oF path. The provided digest mechanisms are for the SCSI family – there is no FC frame level protection evidence and none for NVMe-oF data transfers. Thus, end-to-end digests computed by the migration engine at the source should be able to read in flight and read back at the target, and digests negotiated off for performance will leave transport exposure unaddressed.

Table 6. Validation Layers and Coverage Across Protocol Paths

Layer	FC / SCSI-class path	NVMe-oF path	Residual exposure
Media and controller	Disk-level checks; mismatches still observed in the field [1]	Flash error correction; only post-correction residue is visible [13]	Errors that survive device correction
Transport	Optional iSCSI header and data digests using CRC32C [18]	Not characterized in the corpus	Digests negotiated off; unverified fabric

Layer	FC / SCSI-class path	NVMe-oF path	Residual exposure
Migration-engine digest	CRC32C per block at source read [17]	CRC32C per block at destination read-back [17]	2^{-32} per corrupt block under a random-error model
Cutover gate	Zero mismatches over the full volume	Zero mismatches over the full volume	Clustered corruption missed by sampling alone

Source: Synthesized from the cited records; the corpus contains no frame-level evidence for FC.

4.4 Migration Convergence Under Validation

Validation is related to the dirtying process by migration convergence. Pre-copy transfer: The volume remaining after round i , starting from volume V_0 is described by Equation 5, where the ratio ρ is the dirtying rate D , divided by the effective copy rate B_{eff} . The number of rounds n required to get below the stop-copy value V_{th} is specified by Equation 6, the total time is described by Equation 7, and convergence is obtained with $\rho < 1$ (Equation 8). For network-bound page faults, post-copy with adaptive pre-paging reduced up to 21% network page faults and dynamic self-ballooning avoided free pages transfer [19] which means decreasing the volume being validated is as important as increasing the validation speed. For each of these scenarios, Table 7 and Figure 6 compare the volume's size to 1 TiB and its assumed dirtying rate of 1.8 GiB/s.

$$V_i = V_0 \rho^i, \quad \rho = \frac{D}{B_{\text{eff}}} \quad (5)$$

$$n = \left\lceil \frac{\ln(\frac{V_{\text{th}}}{V_0})}{\ln \rho} \right\rceil \quad (6)$$

$$T_{\text{mig}} = \frac{V_0}{B_{\text{eff}}} \cdot \frac{1 - \rho^{n+1}}{1 - \rho} \quad (7)$$

$$\rho = \frac{D}{B_{\text{eff}}} < 1 \quad (8)$$

Hybrid-copy algorithms use a small pre-copy phase followed by a post-copy phase, and a new variant added a parameter controlling, in real time, when to terminate the iterative copy phase which works well on both read-intensive and write-intensive workloads on Xen [20]. The same applies to the validation criterion: if the stopping criterion is based on the validation adjusted ratio, there are no rounds that cannot converge. As Table 7 indicates, full inline validation at 4 GiB/s does not result in convergence, whereas validation at 16 GiB/s or 10% sampling at 4 GiB/s would cause the convergence point to shift to 1,409 s and 1,033 s, respectively, compared to 853 s without validation. As the dirtying rate gets closer to the effective copy rate, as shown in Figure 7, the total duration diverges.

Table 7. Analytical Convergence Scenarios for a 1 TiB Volume

Scenario	R_h (GiB/s)	f	B_{eff} (GiB/s)	ρ	n	T_{mig} (s)	Downtime (s)
No validation (baseline)	—	—	3.00	0.600	14	853	0.27
Full inline validation, 16 GiB/s	16	1	2.53	0.712	21	1,409	0.33
Full inline validation, 8 GiB/s	8	1	2.18	0.825	37	2,680	0.38
Full inline validation, 4 GiB/s	4	1	1.71	1.050	—	No convergence	—
10% sampled validation, 4 GiB/s	4	0.1	2.79	0.645	16	1,033	0.33

Note: Assumed parameters are $B = 3.0$ GiB/s, $D = 1.8$ GiB/s, and a stop-and-copy threshold of 1 GiB. Values are model outputs, not measurements.

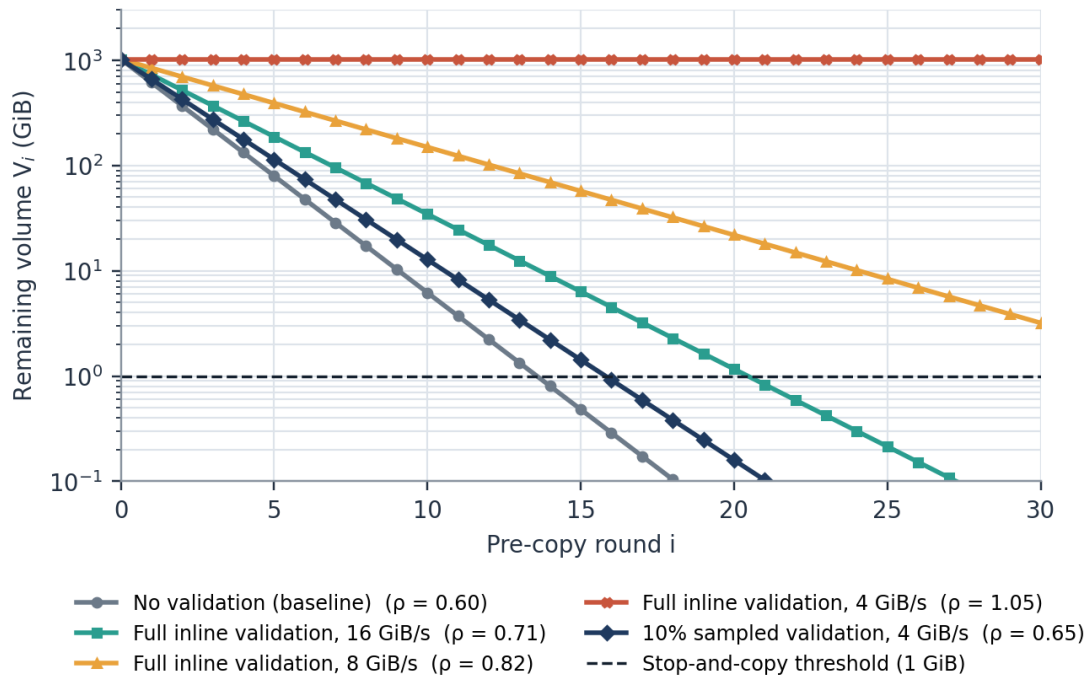


Figure 6. Remaining Volume per Pre-Copy Round Under Validation Scenarios

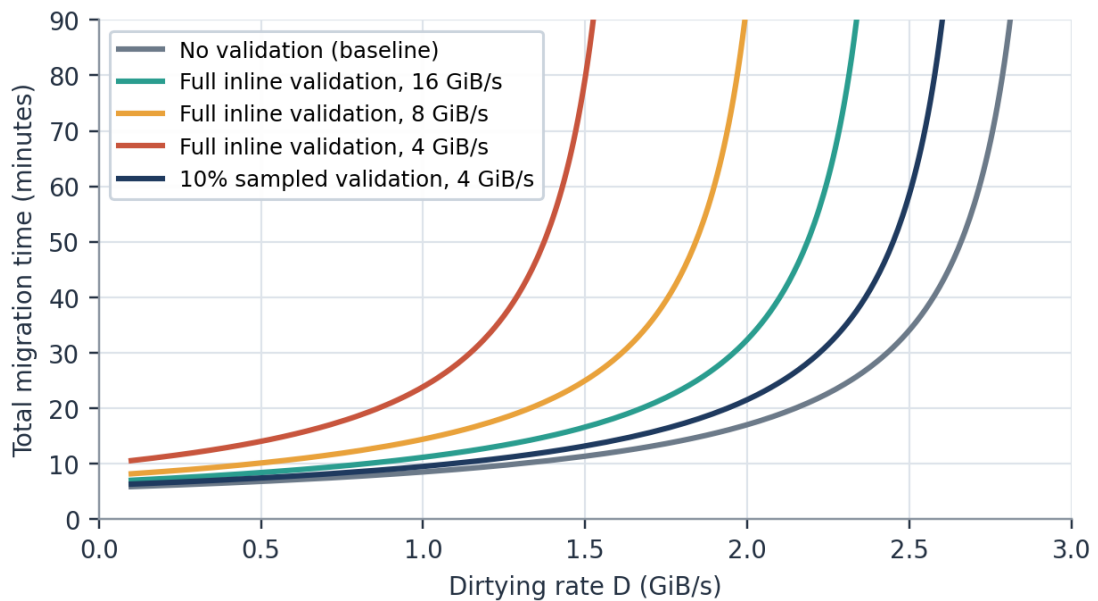


Figure 7. Total Migration Time as a Function of Dirtying Rate

5. CONCLUSION

The systematic review of 20 records looked at how data-integrity validation can be structured when migrating live from Fibre Channel to NVMe storage paths. Field evidence showed that corruption has a rare chance of occurring per disk, but cannot be avoided at fleet scale; with latent errors that had occurred on 3.45% of disks after 32

months, and checksum mismatch errors that had occurred on 0.065% to 0.86% of disks after 41 months, with the errors happening in groups more often than one by one. Assuming that the probability of block corruption is 10^{-7} , digests at the source, in flight and at destination read-back, and a 6.25×10^{-9} undetected corrupt blocks per migrated tebibyte, with a zero-mismatch cutover gate will leave a total of 6.25×10^{-9} undetected

corrupt blocks per migrated teibyte. The cost of that validation, however, controlled the feasibility: full inline hashing at 4 GiB/s resulted in a copy ratio of 1.05 without convergence, while the sampling rate of 10% at the same rate preserved convergence with a copy ratio of approximately 0.65, and a migration time of about 17 minutes. A limited copying strategy, and then a digest comparison of the whole volume, was thus decided on as the most appropriate design for iterative copying.

These conclusions are subject to several qualifications. None of the records

provided had an empirical study of FC-to-NVMe migration and the model values are based upon assumed bandwidth, dirtying and corruption parameters, not measured. The framework should be further validated through instrumented testbeds, fabric-level and frame-level protections be characterized on both paths, identity discrepancies (such as misdirected and lost writes) be detected and fail-slow aware timeouts be adjusted along with the validation fractions so that degraded components are not misinterpreted as corrupted.

REFERENCES

- [1] L. N. Bairavasundaram, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, G. R. Goodson, and B. Schroeder, "An analysis of data corruption in the storage stack," *ACM Trans. Storage*, vol. 4, no. 3, pp. 1–28, 2008, doi: <https://doi.org/10.1145/1416944.1416947>.
- [2] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *IEEE international conference on cloud computing*, Springer, 2009, pp. 254–265. doi: https://doi.org/10.1007/978-3-642-10665-1_23.
- [3] A. Kashyap, S. Guhnani, and X. Lu, "Impact of commodity networks on storage disaggregation with nvme-of," in *International Symposium on Benchmarking, Measuring and Optimization*, Springer, 2020, pp. 41–56. doi: https://doi.org/10.1007/978-3-030-71058-3_3.
- [4] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler, "An analysis of latent sector errors in disk drives," in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2007, pp. 289–300. doi: <https://doi.org/10.1145/1254882.1254917>.
- [5] B. Schroeder, S. Damouras, and P. Gill, "Understanding latent sector errors and how to protect against them," *ACM Trans. storage*, vol. 6, no. 3, pp. 1–23, 2010, doi: <https://doi.org/10.1145/1837915.1837917>.
- [6] V. Prabhakaran, L. N. Bairavasundaram, N. Agrawal, H. S. Gunawi, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "IRON file systems," in *Proceedings of the Twentieth ACM Symposium on Operating Systems Principles*, in SOSP '05. New York, NY, USA: Association for Computing Machinery, 2005, pp. 206–220. doi: [10.1145/1095810.1095830](https://doi.org/10.1145/1095810.1095830).
- [7] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, "A Large-Scale Study of Flash Memory Failures in the Field," in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, in SIGMETRICS '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 177–190. doi: [10.1145/2745844.2745848](https://doi.org/10.1145/2745844.2745848).
- [8] Q. Xu et al., "Performance analysis of NVMe SSDs and their implication on real world databases," in *Proceedings of the 8th ACM International Systems and Storage Conference*, in SYSTOR '15. New York, NY, USA: Association for Computing Machinery, 2015. doi: [10.1145/2757667.2757684](https://doi.org/10.1145/2757667.2757684).
- [9] A. Klimovic, C. Kozyrakis, E. Thereska, B. John, and S. Kumar, "Flash storage disaggregation," in *Proceedings of the Eleventh European Conference on Computer Systems*, in EuroSys '16. New York, NY, USA: Association for Computing Machinery, 2016. doi: [10.1145/2901318.2901337](https://doi.org/10.1145/2901318.2901337).
- [10] M. R. Hines, U. Deshpande, and K. Gopalan, "Post-copy live migration of virtual machines," *SIGOPS Oper. Syst. Rev.*, vol. 43, no. 3, pp. 14–26, 2009, doi: [10.1145/1618525.1618528](https://doi.org/10.1145/1618525.1618528).
- [11] F. Zhang, G. Liu, X. Fu, and R. Yahyapour, "A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 2, pp. 1206–1243, 2018, doi: [10.1109/COMST.2018.2794881](https://doi.org/10.1109/COMST.2018.2794881).
- [12] A. Strunk, "Costs of Virtual Machine Live Migration: A Survey," in *2012 IEEE Eighth World Congress on Services*, 2012, pp. 323–329. doi: [10.1109/SERVICES.2012.23](https://doi.org/10.1109/SERVICES.2012.23).
- [13] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, "Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives," *Proc. IEEE*, vol. 105, no. 9, pp. 1666–1704, 2017, doi: [10.1109/JPROC.2017.2713127](https://doi.org/10.1109/JPROC.2017.2713127).
- [14] Z. Guz, H. (Huan) Li, A. Shayesteh, and V. Balakrishnan, "NVMe-over-fabrics performance characterization and the path to low-overhead flash disaggregation," in *Proceedings of the 10th ACM International Systems and Storage Conference*, in SYSTOR '17. New York, NY, USA: Association for Computing Machinery, 2017. doi: [10.1145/3078468.3078483](https://doi.org/10.1145/3078468.3078483).
- [15] Z. Guz, H. (Huan) Li, A. Shayesteh, and V. Balakrishnan, "Performance Characterization of NVMe-over-Fabrics Storage Disaggregation," *ACM Trans. Storage*, vol. 14, no. 4, pp. 1–18, 2018, doi: [10.1145/3239563](https://doi.org/10.1145/3239563).
- [16] H. S. Gunawi et al., "Fail-Slow at Scale: Evidence of Hardware Performance Faults in Large Production Systems,"

- ACM Trans. Storage*, vol. 14, no. 3, pp. 1–26, 2018, doi: 10.1145/3242086.
- [17] D. Sheinwald, J. Satran, P. Thaler, and V. Cavanna, "Internet protocol small computer system interface (iSCSI) cyclic redundancy check (CRC)/checksum considerations (RFC 3385)," 2002, doi: <https://doi.org/10.17487/RFC3385>.
- [18] M. Chadalapaka, J. Satran, K. Meth, and D. Black, "Internet small computer system interface (iSCSI) protocol (consolidated) (RFC 7143)," 2014, doi: <https://doi.org/10.17487/RFC7143>.
- [19] M. R. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning," in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, in VEE '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 51–60. doi: 10.1145/1508293.1508301.
- [20] Z. Lei, E. Sun, S. Chen, J. Wu, and W. Shen, "A novel hybrid-copy algorithm for live migration of virtual machine," *Futur. Internet*, vol. 9, no. 3, p. 37, 2017, doi: <https://doi.org/10.3390/fi9030037>.