SEVERITY 1 · STORAGE SUBSYSTEM

Storage Array Multi-Device Drop and Recovery

Incident WI-2026-0731-01 · 31 July 2026

ENVIRONMENT	INCIDENT WINDOW	SEVERITY	STATUS
Internal Provider Infrastructure	20:35 to 21:07 local	1 · Critical	Service restored

Executive summary

SUMMARY

On the evening of 31 July 2026, one of the storage arrays at the site stopped working. Four of its twelve hard drives went unreachable at exactly the same moment. That array is built to survive two drives failing, so four was more than it could absorb, and its 26.6 TB of storage went offline. Other storage at the site was unaffected and stayed available throughout, but the services that read from this particular array stopped working.

Service was back thirty-two minutes after Walter Infrastructure (WI) began work on it. Everything on the system was checked and found present and readable. **No evidence of data loss.**

The drives themselves turned out to be fine. The fault was in the electronics that connect the drives to the rest of the machine, which locked up and stopped answering for four of them. Switching the power off and back on cleared it, and all twelve drives have worked normally since, with no errors recorded.

Two points matter going forward. First, the storage came back running on ten of its twelve drives, so it currently has no spare capacity to absorb another failure. Putting the last two back requires several hours of heavy work, which is scheduled rather than rushed. Second, what made the electronics lock up in the first place, after thirty-two days of running without incident, is not yet established. Until it is, WI is treating the machine as unproven.

Because of those two points, WI began copying the highest-value data off that system the same evening, before doing anything else to it. Those copies are running now. The repair work happens once they finish, so that if anything goes wrong during the repair, nothing is lost.

TIME TO RESTORE	DATA LOSS	VOLUME INTEGRITY	DEVICE HEALTH
● 32 minutes	● None found	● Verified, all datasets present	● 12 of 12 nominal
ARRAY REDUNDANCY	**I/O ERRORS SINCE RECOVERY**	**FAILURE MECHANISM**	**TRIGGER**
● 10 of 12, rebuild pending	● Zero	● Identified	● Unexplained, monitored

This report covers Walter Infrastructure's own internal environment, not a client system, and is published as a worked example of the incident process. It is redacted for distribution: host names, network addresses, serial numbers, share names, volume paths and the nature of the stored data have been generalized. Device labels (NAS-01, NAS-02, NAS-03) are used consistently and map to real systems in the unredacted copy held with the site records. Technical detail follows the executive summary.

1. Affected subsystem

SECTION 1

NAS-01 is a sixteen-bay appliance on an ARM SoC platform. Bays are divided between a four-device flash pool and a twelve-device bulk pool. The two pools are independent arrays on independent controller paths, which is material to the analysis: the flash pool was unaffected throughout and remained mounted for the entire event.

ATTRIBUTE	BULK POOL (AFFECTED)	FLASH POOL (UNAFFECTED)
RAID level	`RAID 6, left-symmetric`	`RAID 5`
Member count	`12`	`4`
Chunk size	`64 KiB`	`512 KiB`
Superblock format	`metadata 1.0 (end of device)`	`metadata 1.0`
Usable capacity	`26.6 TB`	`4.7 TB`
Occupancy at incident	`21.6 TB (81%)`	`6.6 GB`
Filesystem	`ext4, data=ordered, delalloc`	`ext4`
Failure tolerance	`2 devices`	`1 device`
Host bus topology	`Single SCSI host, targets 4 through 15`	`4 discrete hosts, one per device`

The twelve bulk devices share one host adapter and are addressed as sequential SCSI targets behind a fan-out stage. The four flash devices each present on their own host. That asymmetry is the first structural clue: a fault confined to one shared stage can take multiple bulk devices without touching flash.

Device ordering. Physical bay numbering on the chassis front panel is offset from the platform's internal slot numbering by the four flash bays. Front bay **N** corresponds to internal slot **N+4**, which corresponds to array device role **N−1**. The mapping was verified in both directions during the response by correlating panel indicators against slot state, and it is recorded here because a misread in either direction during a manual reconstruction would place devices in the wrong stripe order and destroy the array.

2. Scope of impact

SECTION 2

SYSTEM	ROLE	IMPACT	DETAIL
NAS-01 bulk pool	Bulk storage	Offline	Array not assembled. Volume unmounted. Host OS remained up throughout with 32 days uptime.
NAS-01 flash pool	Secondary tier	Nominal	Independent array, independent controller path. Mounted and writable for the entire event.
Consumer A	User facing	Failed	Client-side share mount dead at the OS layer. Confirmed by fresh remount attempt, not a stale session. Container stopped by hand to end a retry loop that had accumulated roughly 4,700 reconnect attempts.
Consumer B	Background	Failed	Two network mounts to the affected pool stale. Stopped during response.
Consumer C	Background	Failed	Shares the same two mounts. Stopped during response.
Hypervisor cluster	7 nodes	No impact	Two storage definitions referencing the array showed inactive on every node. Guest configuration files on all seven nodes were parsed to confirm no virtual disk resided on either definition.
NAS-02, NAS-03	Secondary storage	No impact	Available throughout. Subsequently used as recovery targets.
Backup platform	Guest backups	No impact	Datastore resides on NAS-03 flash tier, physically separate from the affected array. Verified during response.

Dependency discovery is not reliable by name. Three of the four service dependencies were discoverable by searching container mount definitions for the array's identifier. The fourth was not: its storage volume is named for the data it carries rather than the system it comes from, so a name-based search returned a false negative and initially reported it as unaffected. Dependencies were re-enumerated by inspecting volume driver options for the array's network address rather than by matching names. The corrected dependency map is retained with the site documentation.

3. Sequence of events

SECTION 3 · ALL TIMES LOCAL, 31 JULY 2026

TIME	EVENT	TECHNICAL DETAIL
01:03:37	Last coherent array update	Final synchronized superblock write across all surviving members. Established retrospectively from on-disk metadata; not observable at the time.
20:35	Investigation opened	Escalation raised on consumer service failure.
20:37	Fault characterized	Bulk array absent from `/proc/mdstat` entirely. Not degraded, not assembled. Flash pool present and healthy. Host uptime 32 days, confirming a mid-uptime loss rather than a boot-time assembly failure.
20:38	Device enumeration	All 16 block devices present under `/dev`. Platform storage tool flagged four slots and reported the array as requiring recovery, with superblock data reported empty on those four.
20:39	Superblock survey	`mdadm --examine` against all twelve member partitions. Eight returned complete and mutually consistent metadata. Four returned no superblock detected.
20:40	Kernel log analysis	Errors on the four devices all of class `hostbyte=DID_NO_CONNECT`, repeating. Bus addresses correlated.
20:41	Direct read test	`dd` of the first 8 MiB from each raw device. Four failed with I/O error, one of which could not be opened at all. Surviving members returned 94 to 140 MB/s.
20:42	**Failure pattern identified**	Failed devices occupy SCSI targets 8, 10, 12 and 14 on a single host. Targets 4 through 7 wholly unaffected; from target 8 upward, failures alternate exactly. Pattern inconsistent with independent device failure.
20:45	Recovery metadata captured	Full `mdadm --examine` output for all eight surviving members, partition tables, bus topology map and platform storage state written off-device, then committed to two independent repositories.
20:46	Geometry error corrected	Chunk size in prior working notes did not match the captured superblocks. Corrected before any reconstruction was contemplated.
20:47	Controlled shutdown issued	Cold power cycle selected over warm reboot. A warm reboot does not remove power from the fan-out stage and therefore cannot reset it.
20:48:53	Host left network	Expected. Network stack stops early in the shutdown sequence.
20:58	Shutdown stalled	Not complete after 10 minutes. Consistent with the block layer awaiting SCSI command timeouts and error recovery escalation against four unresponsive targets. Forced power removal authorized after confirming the bulk volume was already unmounted, no writes were in flight, and the only mounted filesystem was the near-idle journaled flash pool.
21:00	Cold power cycle	Chassis power removed and restored.
21:07	**Service restored**	Array assembled and volume mounted. State clean, degraded. 10 of 12 devices active, 0 failed, 0 spare.
21:07	Post-recovery verification	All twelve raw devices read nominally. Zero `DID_NO_CONNECT` or I/O error events logged since boot. Volume reports 21.6 TB in use, all datasets enumerable.
21:13	Volume inventory captured	Full directory tree plus recursive long listing with sizes and timestamps. 45,052 file entries, 13,622 directories, written off-device and committed.
21:30	Secondary finding raised	Predictive health warning identified on an unrelated device in NAS-03 during evaluation of that system as a copy target.
22:57	First protective copy started	Detached transfer process launched on NAS-01 itself.
23:03	Priority dataset copy started	Highest-value dataset redirected to the fastest available path.
23:09	All copy legs running	Four datasets, three destinations, one of them multipath.
23:16	Consumer A restored	Verified reading its dataset and self-indexing. Two write-capable consumers deliberately held offline.

On the interval between failure and investigation. The array stopped at 01:03 and was picked up at 20:35. That interval is a function of what the array carries, not of monitoring coverage. No production service, hypervisor node, backup job or other storage array was affected, which is why nothing escalated. The consumers of this array are low-priority services whose brief unavailability carries no operational consequence, and they sit outside the alerting scope deliberately, because alerting on them would bury the alerts that matter. Detection for that service class is by observation rather than by page, and it behaved as designed. Severity is graded on the condition of the infrastructure, not on the criticality of the services it carries: an event that exceeds an array's failure tolerance or carries potential data loss until proven otherwise is Severity 1, and this one did both at detection.

4. Failure mechanism

SECTION 4

Failure mechanism. The SATA fan-out stage that presents the twelve bulk devices to the single host adapter entered a hung state in which it ceased responding for four of its attached targets. The storage devices did not fail and the data recorded on them was never at risk. The condition persisted until power was removed from the backplane. All twelve devices returned to nominal operation. The trigger is a separate question, open and carried in Section 8.

5. Failure signature

SECTION 5

BUS ADDRESS DISTRIBUTION

All twelve bulk devices attach to one SCSI host as targets 4 through 15. Mapping the failures against that address space produced the finding that drove the diagnosis:

```
host:target   state          host:target   state
  4:4         nominal          4:10        NO CONNECT
  4:5         nominal          4:11        nominal
  4:6         nominal          4:12        NO CONNECT
  4:7         nominal          4:13        nominal
  4:8         NO CONNECT       4:14        NO CONNECT
  4:9         nominal          4:15        nominal
```

Targets 4 through 7 are entirely unaffected. From target 8 upward the failures alternate without exception. Four devices failing independently in the same second is already implausible; four devices failing in an arithmetic sequence of bus addresses is not a failure mode that independent devices exhibit. The distribution points at a shared stage, and specifically at a subset of lanes within it.

ERROR CLASS

Every logged error carried `hostbyte=DID_NO_CONNECT` with `driverbyte=DRIVER_OK`. That combination states that the host adapter was unable to reach the target at all. It is categorically different from a device returning a medium error, an unrecovered read error, or a timeout, which are what a failing device produces. The driver layer reported itself healthy while the transport reported no path.

SIMULTANEITY, ESTABLISHED FROM ARRAY METADATA

All eight surviving members carried identical event counters and update timestamps, with a consistent array state bitmap across every member:

FIELD	VALUE ACROSS ALL 8 SURVIVING MEMBERS
Event counter	Identical on all eight
Update timestamp	Identical on all eight, 01:03:37
Array state bitmap	Consistent, showing the same four positions absent
Checksum status	Correct on all eight
Device role assignment	Sequential and complete for the surviving set

Identical event counters across every survivor establish that the array stopped at a single instant with no split-brain condition and no torn writes. Progressive device degradation produces divergent event counters as members drop at different times. This state is the strongest single evidence of a bus event, and it is what made recovery low risk: the surviving stripe set was internally coherent.

5. Failure signature, continued

SECTION 5

PHYSICAL PRESENCE MAINTAINED

All sixteen block devices remained enumerated under `/dev` for the duration; nothing disappeared from the bus. The devices were addressable as objects while unreachable for I/O, consistent with a fan-out stage that has stopped forwarding transactions rather than one that has dropped its targets.

RECOVERY BEHAVIOR

A cold power cycle restored all four devices to full function; a genuinely failed storage device rarely does, while a hung controller routinely does. With the address pattern and the coherent surviving metadata, the evidence is substantially more consistent with a hung shared transport stage than with four independent device failures. Confirmatory rather than diagnostic, but it weighs heavily against device failure.

6. Post-recovery array state

SECTION 6

The array reassembled with ten of twelve members. Understanding why two rejoined and two did not is central to the remediation plan, because the two categories require different handling.

DEVICE GROUP	COUNT	SUPERBLOCK STATE	OUTCOME ON RESTART
Never affected	8	`Intact, coherent`	Assembled normally.
Affected, metadata intact	2	`Intact, readable once bus restored`	Rejoined automatically and are active, in-sync members. Their metadata was never damaged; it was unreadable through the hung stage, which the platform tool reported as an empty superblock.
Affected, metadata absent	2	`Genuinely absent`	Did not rejoin. Read nominally at the device layer but hold no array metadata, so they cannot be reassembled and must be added as new members.

> **An early reading was corrected here.** The platform's storage tool reported empty superblocks on all four affected devices during the fault. That reading was accurate for two of them and an artifact of unreadability for the other two. The distinction was not determinable while the bus was hung, and it materially changed the recovery outlook: two devices returning with intact metadata is the difference between a ten-member assembly and an eight-member one, and an eight-member assembly of a twelve-device RAID 6 does not mount.

CONSEQUENCE FOR REDUNDANCY

A twelve-device RAID 6 sustains two absent members. The array currently has exactly two absent. It is therefore operating correctly but with zero remaining tolerance: a single further device event during the present window would take the volume. This is the dominant residual risk and it governs the sequencing in Section 11.

7. Stale device error flags

SECTION 7

Four bay indicators continued to show error state after recovery, including two bays whose devices were at that moment active, in-sync array members. Inspection of the platform configuration store found exactly four persisted per-device error keys, one per illuminated bay.

PROPERTY	FINDING
Flag count	`4, matching the four illuminated bays exactly`
Storage location	Platform configuration store, persisted across reboot
Correlation with live state	None. Two flagged bays are active array members.
Correlation with device health	None. All four devices test nominal.
Clearing method	Removal of the persisted keys followed by a restart
Operational significance	Cosmetic, but actively misleading during triage

These indicators cannot be used to assess array health on this platform. During the response the panel reported four failures while the actual state was zero failed devices and two non-member devices. Any future triage on this hardware should treat bay indicators as advisory only and derive state from the kernel array table and direct device reads.

8. Trigger analysis

SECTION 8 · UNRESOLVED

The mechanism is established. The trigger is not. The unit had been running thirty-two days without incident and had not been physically disturbed, opened, or moved for several years prior. The following remain open and are ordered by assessed likelihood:

CANDIDATE	ASSESSMENT
Controller firmware defect	Consistent with a hang that clears on power cycle and leaves no device-level evidence. A firmware revision review is pending and is the cheapest possible mitigation if a relevant fix exists.
Transient power condition	Would plausibly affect a subset of lanes on a shared stage. No corroborating evidence available; the platform does not log power events at the resolution required.
Thermal event	Device temperatures sampled post-recovery are nominal. No historical thermal record retained at useful granularity, so this cannot be excluded retrospectively.
Backplane degradation	Considered and rated unlikely. A physical connection fault does not self-repair across a power cycle. Not excluded entirely, since intermittent contact under thermal cycling could present this way, but it does not fit the evidence as well as the alternatives.

9. Hypotheses raised and tested

SECTION 9

Five hypotheses were formed during the response and none survived testing. The verdict on each is worded to match the strength of the test behind it. They are recorded in full because the eliminations are part of the finding, and because two of them were held with high confidence before being tested.

H1 · FOUR DEVICES FAILED INDEPENDENTLY

Rejected on the evidence. All four returned to full function after power cycling and subsequently tested indistinguishably from their peers on sequential read and seek latency. Four independent failures do not distribute across an arithmetic sequence of bus addresses, and the simultaneous return of all four on a single power cycle is not a behavior independent failures produce.

H2 · BACKPLANE OR CABLING FAULT

Substantially less likely. This was the leading hypothesis through the first phase of the response and it did not survive the recovery. The alternating-lane distribution fit a physical connector or lane fault well, and hardware replacement was being scoped on that basis before the cold cycle. A fixed connection fault does not repair itself when power is removed and restored; an intermittent contact fault could, so this one is downgraded rather than closed, and Section 8 retains backplane degradation as an open, lower-ranked candidate.

H3 · ARRAY METADATA CORRUPTION

Disproved by direct observation. Metadata reported absent on four devices during the fault proved intact on two of them once the bus recovered. Superblocks in this format reside at the end of the member partition; a read of that region is still a read, and an unreachable device returns the same result as an unwritten one. The platform tool did not distinguish the two cases.

H4 · ARRAY WRITE PERFORMANCE LIMITED BY STRIPE CACHE DEPTH

Disproved by measurement. During post-incident performance analysis of NAS-03, the RAID stripe cache was found configured at half the kernel default. On a parity array a starved stripe cache prevents write coalescing into full stripes and forces read-modify-write cycles, which fit the observed write throughput closely. The parameter was raised through four values spanning a 64-fold increase and measured at each step.

STRIPE CACHE DEPTH	SUSTAINED WRITE
Baseline (kernel default halved)	96.9 MB/s
8x baseline	93.6 MB/s
32x baseline	95.1 MB/s
64x baseline	92.5 MB/s

Flat within measurement noise. The hypothesis was rejected and the parameter returned to its original value rather than left changed on the strength of a theory.

H5 · ARRAY WRITE PERFORMANCE LIMITED BY FILESYSTEM STRIPE MISALIGNMENT

Disproved by measurement, but a real defect was found and corrected. The ext4 stripe width recorded on the affected volume corresponded to a smaller member count than the array actually has, consistent with the array having been expanded at some point without the filesystem geometry hint being updated. Every write batch was therefore being aligned to a boundary that did not match the true stripe, which forces read-modify-write on a parity array and fit the symptom precisely.

The geometry was corrected to match the array. Throughput before and after was measured across three runs and did not move. The misconfiguration was genuine and worth fixing on its own merits, but it was not the constraint. Both H4 and H5 were plausible, both were measured rather than assumed, and both were discarded when the data did not support them.

> **Why rejected hypotheses appear in this report.** An RCA that lists only the conclusion gives the reader no way to judge how well the conclusion is supported. Recording what was tested and rejected distinguishes what is known from what was expected, and prevents a future engineer from re-running the same eliminated tests during the next event on this hardware.

10. Method

SECTION 10

The array was recoverable or it was not, and that could not be known until it was characterized. Sequencing every step to preserve the means of recovery, before attempting any recovery, followed from that.

READ-ONLY UNTIL CHARACTERIZED

No corrective action was taken until the failure mode was understood. The diagnostic set comprised device enumeration, bus topology mapping, kernel ring buffer analysis, array superblock inspection across all members, raw device reads, and platform storage state queries. None of these writes to the array. A rebuild initiated at this stage against the device-failure hypothesis, which was the initial reading, would have written parity across a coherent stripe set on the basis of a misdiagnosis.

METADATA CAPTURE PRECEDING ANY STATE CHANGE

Complete superblock output for every surviving member, plus partition geometry, bus topology and platform storage state, was written off the device and committed to two independent repositories before the shutdown command was issued. That capture is what would have permitted a manual array reconstruction had assembly failed on restart, since it preserves device role ordering, chunk size, metadata format and array identity.

The capture also caught an error. A geometry parameter recorded in prior working notes did not match the value in the captured superblocks. Reconstruction using the incorrect value would have written a stripe layout inconsistent with the on-disk data and destroyed the volume irrecoverably. The corrected value was propagated to the documentation before any recovery path was contemplated.

COLD CYCLE SELECTED DELIBERATELY OVER WARM RESTART

A warm restart does not remove power from the storage backplane or the fan-out stage, so against a transport-level hang it changes nothing while consuming the one clean shutdown state available. Full power removal was selected specifically because the diagnosis placed the fault below the device layer.

When the shutdown stalled past ten minutes, forced power removal was authorized only after establishing three conditions: the bulk volume was already unmounted so no writes were in flight, the surviving members held coherent metadata so no in-progress stripe update could be truncated, and the only mounted filesystem was the near-idle journaled flash pool.

INVENTORY PRECEDING PROTECTION

A full recursive listing of the volume was captured while mounted and before any copy operation began: 45,052 file entries with sizes and timestamps, and 13,622 directories. Had the array been lost during the subsequent multi-hour copy, that listing would have been the authoritative record of contents.

A first attempt at the inventory silently produced directories but no file entries, since the platform's embedded userland lacks the printf extension used. The failure was caught by validating the output row count against expected magnitude rather than by assuming success, and the method was changed to a recursive long listing.

> **No changes were made to the affected array during diagnosis.** No rebuild, no device removal or reseat, no array creation, no writes. The only state change to NAS-01 during the incident window was the power cycle itself.

11. Verification performed

SECTION 11

CHECK	RESULT	MEASURE
Array assembly state	Degraded	Active on 10 of 12, 0 failed, 0 spare, state clean
Volume mount and enumeration	Pass	Mounted, 21.6 TB in use, all datasets readable
Raw device read, all 12	Pass	148 to 264 MB/s sequential, sampled at three offsets each
Seek latency, all 12	Pass	Within a 7% band across the set, no outliers
Kernel error count since boot	Pass	Zero transport or I/O errors
Array sequential read, degraded	Pass	560 MB/s sustained
Guest storage exposure	Pass	All 7 hypervisor nodes parsed, no guest disk on either storage definition
Backup datastore integrity	Pass	Resides on separate hardware, unaffected
Device health telemetry	Unavailable	Platform health tooling returns no usable output on this model. Device condition currently inferred from performance testing. See Section 15.

12. Transport selection

SECTION 12

With the array carrying no remaining tolerance and the trigger not yet established, protective copies were started ahead of the rebuild. Path and protocol selection were driven by measurement on the real path rather than by specification, and several intuitive assumptions did not survive testing.

PATH OR TEST	RESULT	FINDING
NAS-01 degraded array, sequential read	560 MB/s	RAID 6 dual-parity reconstruction cost on this SoC is far lower than assumed
NAS-01 to NAS-03 disk tier, local benchmark	93 to 97 MB/s	Did not predict network transfer rate. See below.
NAS-01 to NAS-03 disk tier, 80 GB sustained with fsync	235 MB/s	2.5x the local figure. The local benchmark is not a valid predictor for this workload.
NAS-01 to NAS-03 flash tier	367 MB/s	Single large sequential file
NAS-01 to NAS-02, single path	70.7 MB/s	Link limited. Target writes at 407 MB/s locally.
Alternative transport, same path	166.9 MB/s	Against 367 MB/s for the selected protocol. Processor bound on the ARM SoC.
Many-small-file workload, flash tier	42 to 95 MB/s	Same path that benchmarks at 367 MB/s on one large file

BENCHMARK VALIDITY

Three measurement traps were identified and controlled for. Each would have produced a materially wrong plan.

TRAP	CONTROL APPLIED
Write cache flattering short tests	Targets export with asynchronous write semantics and hold roughly 10 GB of free memory, so a 4 GB write reports near-memory speed. All planning figures were taken from 80 GB transfers with an explicit sync, well past cache.
Block size against stripe geometry	The NAS-03 disk array is a five-member RAID 5 with a 512 KiB chunk, giving a 2 MiB full stripe. A 1 MiB test block forces read-modify-write on every operation and produced 19 to 21 MB/s, roughly a fifth of the stripe-aligned figure of 93 MB/s. All benchmarks were re-run at or above full-stripe width.
Virtual interface speed reporting	Bridge and firewall-bridge interfaces report a nominal 10 Gb/s regardless of the physical link beneath them. Link speeds were taken from physical interface members only, which corrected an initial overestimate on one node by a factor of ten.

PROTOCOL SELECTION

Two transports were measured on identical hardware and path. The selected protocol delivered 2.2 times the throughput of the alternative, with the difference attributable to per-byte processor cost on the ARM SoC rather than to the network. With four transfer legs running concurrently against the same processor, the lower-overhead protocol was selected for all legs rather than mixed, so that no leg competes for cycles with the others.

13. Multipath transfer engineering

SECTION 13

NAS-02 carries the largest transfer leg and sits behind a 1 Gb/s link, making it the critical path for total elapsed time. Two options were evaluated to raise its throughput.

LINK AGGREGATION, EVALUATED AND REJECTED

The core switch supports 802.3ad and exposes link aggregation group configuration, membership, and per-port protocol parameters. It exposes no load-balancing or hash policy setting at any level of its configuration interface. With a fixed firmware hash, every frame between one source address and one destination address resolves to the same physical member, so a single transfer between two hosts would occupy one link and gain nothing. Aggregation was rejected on that basis before any cabling was changed.

DISCRETE-ADDRESS MULTIPATH, IMPLEMENTED

The second network interface on NAS-02 was assigned its own address in the same subnet, and the transfer was split into two independent streams targeting the two addresses. Distinct destination addresses on distinct interfaces produce genuinely distinct paths with no dependence on switch hash behavior.

The known failure mode for same-subnet multi-homing is the host answering address resolution for either address on either interface, which collapses both streams onto one link. Address resolution behavior was constrained so that each interface responds only for its own address, and the result was verified from the source host rather than assumed: the two destination addresses resolve to two different hardware addresses, matching the two physical interfaces.

> **The second interface was found carrying a stale configuration** from a previous network topology, on a subnet with no route anywhere in the current environment. It had link state up and would have appeared functional to a casual check, while being unreachable from every host. It was reconfigured as part of this work.

14. Copy execution design

SECTION 14

Transfers use `rsync` over NFS, driven from the shell on NAS-01 itself rather than through the appliance's own replication interface. The source host holds local read access to every dataset, whereas an intermediary would require additional exports for two of the four datasets and would introduce a second network traversal. The source host is also the only component whose failure would halt the work regardless of where it was orchestrated, so placing the orchestrator there adds no failure mode.

WHY THE COMMAND LINE RATHER THAN THE APPLIANCE'S OWN SYNC TOOLING

The appliance ships backup and replication tooling that would have moved this data. It was not used, because this work had four requirements that are more readily met by driving `rsync` directly.

REQUIREMENT	WHY IT MATTERED HERE
Explicit phase sequencing	Which datasets run, in what order, and how many concurrently was the single decision governing total elapsed time. Running all four at once cost roughly 40% of achievable throughput to seek contention on the source array. That ordering needed to be expressed directly rather than configured per job.
Append-mode resume on one very large object	The archival object is 1.5 TB. Checksum-verified resume re-reads the copied prefix on every restart, which is correct for many-object datasets but grows without bound on a single large one. Selecting append mode for that one transfer and verified mode for the others required per-transfer control of the resume semantics.
Plain-text logs outside the appliance	Progress and failures needed to be readable from any host, and retained afterward for this report, without depending on the appliance's own interface being reachable.
Bounded mount-health checking	A hung network mount blocks indefinitely. The transfer wrapper tests mount responsiveness with a timeout and forces a remount rather than waiting, which is custom logic around each attempt.

The trade is that the work is scripted rather than configured, which is the correct trade for a sequence that runs unattended against an array with no redundancy. This is a selection on fit, not a judgment on the vendor tooling, which is appropriate for routine scheduled replication.

DESIGN ELEMENT	IMPLEMENTATION AND RATIONALE
Process detachment	Session-detached processes. Verified by test rather than assumed: a job was started, the control session was closed, and the job was confirmed still progressing on reconnection. The platform's terminal multiplexer is present but non-functional for detached launch on this build, which the test caught.
Resumability	Partial transfers retained and resumed by append with checksum verification of the existing prefix, so an interruption mid-file resumes rather than restarting a multi-hundred-gigabyte object.
Failure handling	Each leg wraps its transfer in a retry loop that re-establishes its mount before each attempt. A transport interruption pauses the leg rather than ending it.
Ownership handling	Owner and group preservation deliberately omitted. Destination exports squash to a fixed identity, so preservation would emit an error per object. Destination ownership was aligned to the squash identity so that timestamp preservation succeeds and the transfer can terminate with a success code rather than looping indefinitely on a benign permission error.
Logging	Per-leg logs written to the unaffected flash pool, not to the degraded array.
Destination isolation	Dedicated destination paths created outside the platform's managed share structure, so no destination is visible to any indexing service on the target appliances. Writing into an existing indexed path would have produced duplicate catalog entries.
Service quiescing	Two write-capable services held offline for the duration. One writes sidecar files into the source tree and the other rewrites stored objects in place. Either would mutate source data mid-copy and render the copy inconsistent with its inventory.

DATASET	DESTINATION	PATH	RATIONALE
Dataset A	NAS-03 flash tier	Single	Highest object count of the four. Flash absorbs per-object overhead far better than a parity disk array.
Dataset B	NAS-03 disk tier	Single	One large sequential object, best case for a parity disk array.
Dataset C	NAS-03 disk tier	Single	Sequenced behind Dataset B on the same leg to avoid seek contention on one array.
Dataset D	NAS-02	**Dual**	Split across two physical paths. Placing the larger of the two bulk datasets on the faster link reduces total elapsed time by roughly 15 hours against the inverse assignment.

15. Current state

SECTION 15

ITEM	STATE	DETAIL
Volume availability	**Available**	Mounted and serving
Array redundancy	**None**	10 of 12 members against a tolerance of 2
Device health	**Nominal**	All 12 reading at expected rates
Transport errors since recovery	**Zero**	Clean since 21:07
Protective copies	**Running**	Four datasets, three destinations, no errors logged
Consumer A	**Restored**	Verified serving and self-indexing
Write-capable services	**Held**	Deliberately offline pending copy completion
Bay indicators	**Stale**	4 false errors, clearing bundled with the rebuild window
Device health telemetry	**Unavailable**	No working method on this platform, see Section 17

16. NAS-03 predictive device warning

SECTION 16 · UNRELATED TO THE INCIDENT

Identified while evaluating NAS-03 as a copy destination. One device in its five-member parity array carries a predictive health warning from platform monitoring. The array is at full redundancy and the device performs normally under test.

ITEM	STATE	DETAIL
Array redundancy	**Full**	All 5 members active and synchronized
Flagged device, sequential read	**Nominal**	Within the range of its peers at three sampled offsets
Flagged device, seek latency	**Nominal**	Mid-range for the set. The slowest device in the array is a different member.
Device temperature	**Nominal**	No thermal contribution
Warranty	Outside coverage term, so replacement proceeds as a purchase rather than a claim.	
Replacement	**Scheduled**	Specified and scheduled as planned lifecycle work. The array holds full redundancy throughout, and the device performs within its peer range under test.
Peer devices	**Watch**	Two further members are the same model and probable production batch, at equal age under equal workload

Do not remove the flagged device ahead of its replacement arriving. The array tolerates one absent member. Removing a functioning device early converts a fully redundant array into a degraded one for the entire procurement lead time, which inverts the intent. The correct sequence is to hot swap the replacement while redundancy is intact so that the rebuild proceeds from a healthy state.

A related benchmarking artifact was identified on this array. Its write throughput initially measured at roughly a fifth of its true rate because the test block size was half the array's full stripe width, forcing read-modify-write on every operation. The array is not fast, but it is not as slow as the first measurement indicated. Any future performance assessment of this array must use stripe-aligned block sizes.

17. Backup platform verification

SECTION 17

The backup platform was examined during the response to establish whether the incident placed guest backups at risk. It did not. The datastore resides on NAS-03's flash tier, a different array on different hardware from the affected pool, and was available and intact throughout.

PROPERTY	FINDING
Datastore availability	Unaffected for the duration of the incident
Deduplication ratio	4.88 to 1, logical against physical. Material to any retention change: reducing retained snapshot count reclaims substantially less space than the count reduction implies, and reclamation occurs at garbage collection rather than at prune time.
Server rebuild path	Datastore content is fully recoverable by a replacement server pointed at the existing store, verified during the response. A lost backup server therefore costs configuration time rather than data, which lowers the criticality of protecting the server itself.
Coverage, current guests	Current to the most recent scheduled run
Coverage, recently provisioned guests	Two guests have no snapshot history. Both were provisioned after the current job definition was written and have not yet reached their first scheduled run, which is expected rather than a failure.
Retention review	Four guests hold snapshots older than the others. Scheduled for review alongside the retention policy, which is being revisited in light of the deduplication figure above.

The scheduled job is functioning. Coverage was verified against the platform's own guest inventory rather than assumed from job status, which is what surfaced the two pending guests and the retention question. Both are carried in Section 20.

18. Device health telemetry gap

SECTION 18

Neither standard device health tooling nor the platform's own health utility returns usable output on the NAS-01 hardware. Standard tooling is absent from the firmware image and the vendor utility returns empty results for every queried slot, including slots whose devices are demonstrably healthy. Device condition on this array is therefore currently inferred from throughput and latency testing rather than read from the devices.

This is a material limitation. A declaration that the array is healthy cannot presently be made on evidence, only on the absence of contrary indication. Resolving it is a prerequisite for closing the incident.

19. Actions completed

SECTION 19

ACTION	STATUS	OUTCOME
Recovery metadata captured off-device	**Complete**	Committed to two repositories
Array geometry error corrected	**Complete**	Documentation now matches on-disk state
Cold power cycle	**Complete**	Fan-out stage cleared, 12 of 12 devices returned
Volume integrity verified	**Complete**	All datasets present and readable, no evidence of loss
Full volume inventory captured	**Complete**	45,052 files, 13,622 directories, held off-device
Guest storage exposure audited	**Complete**	7 nodes, no exposure
Service dependency map corrected	**Complete**	Four consumers, one recovered from a false negative
Destination paths provisioned	**Complete**	Isolated from managed share structure on two systems
Transport measured and selected	**Complete**	2.2x throughput over the alternative
Multipath provisioned and verified	**Complete**	Distinct hardware addresses confirmed from the source host
Stale interface configuration corrected	**Complete**	Second NAS-02 interface returned to the routed network
Protective copies started	**Running**	Four datasets, three destinations
Consumer A restored	**Complete**	Verified serving
Write-capable services quiesced	**Deliberate**	Held pending copy completion

20. Pending actions

SECTION 20

CRITICAL **Restore array redundancy.** Add the two non-member devices and rebuild to full membership. They hold no array metadata and therefore join as new members rather than rejoining, requiring a full parity reconstruction. **Window required.** Several hours at sustained load, and it is the heaviest I/O the array will see. Must not start until protective copies complete.

HIGH **Instrument for recurrence.** Monitoring for the specific transport error class has been specified and should be wired to the existing alert path. A single recurrence following a clean rebuild changes the platform decision from repair to replace.

HIGH **Resolve device health telemetry.** Prerequisite for declaring the array healthy on evidence rather than on absence of symptoms. See Section 18.

HIGH **Review controller firmware.** Establish whether a revision exists addressing fan-out or transport behavior. Simplest available mitigation if one exists. Complete before the rebuild window.

HIGH **Clear stale bay indicators.** Requires a restart. Bundle into the rebuild window rather than consuming a separate outage.

MEDIUM **Replace the flagged NAS-03 device.** Procure, then hot swap while redundancy is intact. **Window required** for the rebuild that follows.

MEDIUM **Extend the protection tier for the two highest-value datasets.** Both are held on dual-parity storage and were never at risk during this event. Adding a second independent tier is scheduled work, brought forward as a direct output of this incident.

MEDIUM **Confirm first scheduled backup of the two newly provisioned guests** and complete the retention policy review. See Section 17.

LOW **Persist the multipath interface configuration.** Currently applied at runtime and will not survive a restart of that host, which would stall one copy leg into indefinite retry.

21. Open questions

SECTION 21

QUESTION	BEARING
What triggered the fan-out stage to hang after 32 days of nominal operation?	Determines repair against replace. Firmware, transient power and thermal causes all remain open, ranked in Section 8.
Does a firmware revision exist addressing this behavior?	Cheapest available mitigation. To be answered before the rebuild window.
Can device health telemetry be obtained on this platform by any method?	Without it, array health is inferential. Affects whether this hardware remains suitable for this role.
Do the two remaining same-batch devices in NAS-03 warrant proactive replacement?	Assessed against remaining service life and the array's redundancy position.
Should NAS-01 be replaced rather than rebuilt?	Both paths are documented. The decision gate is recurrence, which the instrumentation above is designed to answer. The existing devices are healthy and portable to another chassis, so replacement cost is the enclosure rather than the storage.

22. Assessment

SECTION 22

Contained with no evidence of data loss, restored in thirty-two minutes, the failure mechanism identified to the subsystem. The affected hardware is operating within normal parameters and has logged no errors since recovery.

The platform is not yet proven and is being treated accordingly. Redundancy restoration is scheduled rather than assumed, recurrence instrumentation is specified, health telemetry is flagged as a gap rather than glossed, and the highest-value datasets are being copied off the array while the trigger question stays open. Two findings unrelated to the incident were surfaced by the response and are carried above with priorities.

Dave Walter

Walter Infrastructure · Principal Engineer

Report issued: 1 August 2026

Next update: on completion of protective copies

> **Sanitized example.** Prepared from the incident record for the named environment. Identifiers generalized for distribution; the unredacted record is retained with the site documentation.