

When milliseconds turn into seconds.

The impact of cloud storage performance on digital preservation operations.

Library of Congress – Designing Storage Architectures, 2026

Merritt

California Digital Library's
repository by the numbers.

- 101.4 million files
- 622.4 TB primary copy
- 4.7 million objects
- 1.86 PB for 3 copies total; managed at the file level
- 14 years

29 repository system and microservice functions checked on a daily basis.



SDSC



Amazon S3



Fixity Check Service

- Typical cycle time for 304 million files is roughly 50-55 days
- Glacier: File header checks only
- For all other files, we calculate fixity on the byte stream to optimize performance.

Fixity Check Status Criteria

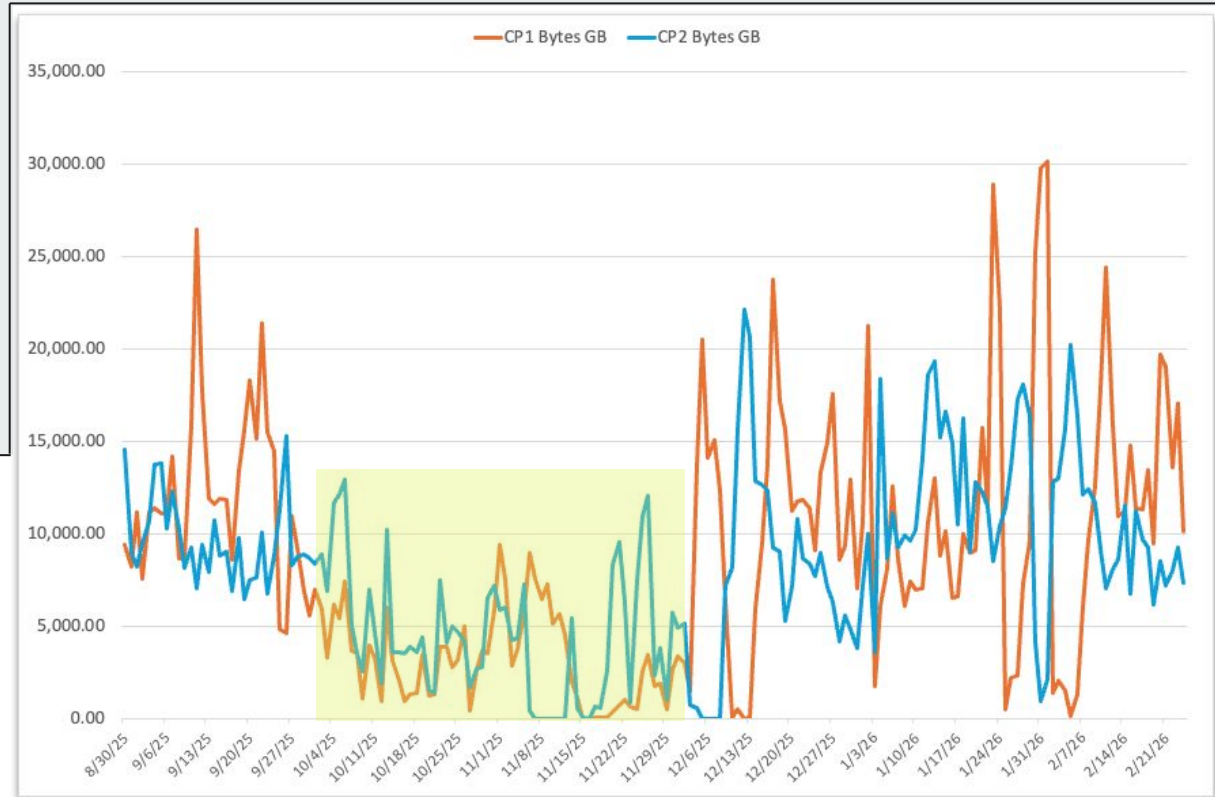
**Beginning in early October 2025,
we noticed that our criteria for
bytes audited in 24hrs was no
longer being met.**

PASS Process more than 2.3m files AND more than 12 TB of online content

INFO Process more than 1.55m files AND more than 8 TB of online content

FAIL Process less than 1.55m files AND less than 8 TB of online content

Audit bytes processed



Spoiler

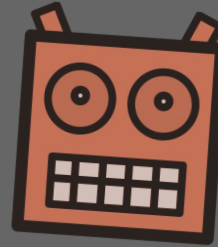
A long-lived, but ultimately transient performance issue at Cloud Provider 2 (CP2) masked the real issue at CP1.

From early October through the end of November, Audit performance criteria were not met on most days.

At this point, we saw our expected fixity check cycle time nearly double.

No cloud service provider outages and no planned maintenance announcements.

We started thinking about everything we could test...



RDS parameters; **Additional, robust debugging to Audit service**; Investigate EBS volume bursts; **Audit process isolation on a per cloud provider basis**; Add JVM garbage collection changes; **Modify Audit Filters to focus only on new content**; Lower Audit batch size; **Isolate failures on API requests**; CP1 outbound routing change; **AWS suggests more powerful EC2 instances**; Toggle AWS API v2 in Merritt production; **Possible trial of API v2 explicit streaming feature that could help in the future (testing indicates this method is slower)**; V1/V2 performance comparison (similar averages in V2 and V2); **Benchmark handling of a 100MB file with CP1 storage**; Lower application thread count in case pending S3 api calls are interfering with each other; **Raise application thread count**; Set filters to only process only large files; **Configure each Audit host to handle only one cloud provider**; Halt all audit processing for X hours and re-evaluate download times; **Ask cloud providers to repeat presigned download tests to evaluate for swings in performance.**

Gaining focus

Adopted a benchmark test matrix

- Access timing x 3 CPs - using pre-signed URLs
- Audit timing x 3 CPs - using byte stream duration
- Glacier excluded

Ad hoc usage of this matrix detected a sustained degradation of performance with CP1 in November.

60MB file downloads were consistently taking 50-65s.

Based on benchmarks, this was 36x slower than where we wanted to be.

CP1 investigates

- *Uplink to AWS*
- *Distribution switch groups*
- *Uplinks to switch groups*
- *Load balancers*
- *Network traces*
- *Internal timing tests*

Network traces of GET requests showed retransmissions and packets being received out of order.

Ultimately, hosts were moved to newer switching infrastructure to take advantage of new load balancers.

Success!

In early December, our download and Audit tests showed drastic improvement at CP1.

Min time: 998ms

Max: 1.456 sec for 60MB

```
node_number: "9501",
filename: "100_mb_random.dat",
...
audit_expected_retrieval_time_sec: 3.1165824,
cli_command:
  "aws s3 --profile sdsc --endpoint-url https://cdl.s3.xxx:443 cp
  s3://cdl.xxx.prod/ark:/99999/fk4r22r39v|1|producer/100_mb_random.
  dat /dev/null",
results: {
fixity_status: "verified",
retrieval_time_sec: 1.7569144170158543,
status: "PASS"
```

Additional measures taken

Added the ability to log low-level timing data in ms in the Audit service.

timeMS

Time from the start of an audit operation for a file until its completion which includes a database update

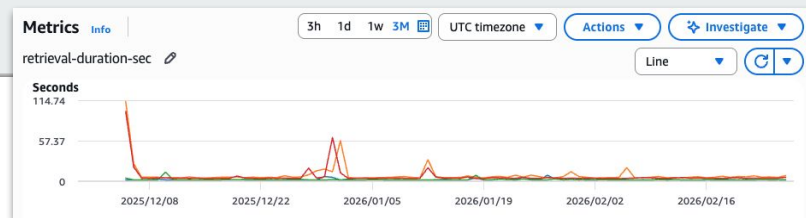
streamMS

Time for obtaining the input stream + the calculation time for the file digest

Captured in OpenSearch

```
"cdl": {...
  },
  "@version": "1",
  "json": {
    ...
    "batchStat": {
      "nodeInfo": {
        "nd-5001": {
          "bytes": 147707,
          "timeMs": 1177,
          "count": 18,
          "streamMs": 872
        }
      }
    }
  }
}
```

Since then...



On five occasions, we've observed service slowdowns with CP2, akin to when it masked CP1's issue earlier. Duration varied from 2-7 days.

- The maximum deterioration in performance resulted in Audit and access responses that were up to 16x slower than expected: 113s vs. 7s for 100MB
- The minimum deterioration resulted in responses that were roughly 2x slower than expected.
- During these windows, average performance was 6.6x slower than expected.

Conclusion

All this goes to show

- Problems can be masked when working across multiple cloud SPs.
- Hidden hardware issues can directly impact even distributed, multi-host processing and eventually impact SLAs.
- Even regular, "business as usual" deterioration in performance can have a direct, negative effect on core digital preservation processes.

Be vigilant, establish metrics and collaborate!

Thank you! Questions?

merritt.cdlib.org/presentations

Eric Lopatin

eric dot lopatin at ucop dot edu

University of California Curation Center – UC3 