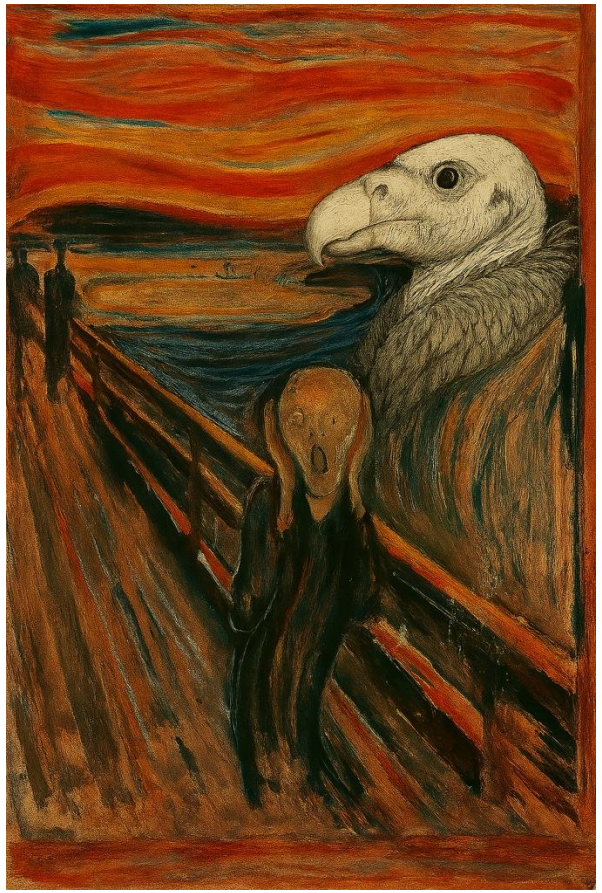
Unbreaking the bird: Debugging Pelican client failures

condor_q -held | grep osdf



An AI-generated nightmare courtesy of Copilot.

So you walked into the office the morning and checked on the workloads that ran overnight.

Oh no! It looks like OSDF caused all sorts of errors overnight.

What do you do?!

N.b. – we're on your side! Every morning we check in on how many problems there were caused by OSDF on the previous day…

Per OSDF endpoint download (i.e. input transfer) statistics

Endpoint Name	Endpoint Institution	Total Attempts	Total Jobs	Successful Attempts	Successful Jobs	Failed Attempts	Pct Attempts Failed	Failed Attempts per Job	Num Jobs w/ Failed Attempts	Num Jobs Interrupted	Pct Jobs Interrupted
TOTALS		731,926	704,066	717,015	704,030	14,911	2.0%	0.02	1,487	1,14	0.2%
MGHPCC_NRP_OSDF_CACHE	Massachusetts Green High Performance Computing Center	234,469	234,319	234,411	234,307	58	0.0%	0.00	56	2	0.0%
CHTC_PELICAN_CACHE	University of Wisconsin–Madison	117,061	114,209	113,088	113,119	3,973	3.4%	0.03	1,015	294	0.3%
BOISE_INTERNET2_OSDF_CACHE	Internet2	106,785	106,192	106,772	106,183	13	0.0%	0.00	13	1	0.0%
JACKSONVILLE_INTERNET2_OSDF_CACHE	Internet2	97,903	95,601	97,846	95,568	57	0.1%	0.00	55	4	0.0%
Stashcache-Kansas	Internet2	87,728	83,374	82,934	82,271	4,794	5.5%	0.06	1,189	942	1.1%
Stashcache-Chicago	Internet2	30,689	26,969	26,661	25,935	4,028	13.1%	0.15	1,055	642	2.4%
Sunnyvale-I2-PRP	Unmapped endpoint Sunnyvale-I2-PRP	17,580	17,573	17,568	17,561	12	0.1%	0.00	12	0	0.0%

So your Pelican-power download failed…

12635414.5 XXXXXXXXX 5/6 10:46 Transfer output files failure at execution point slot1_1@glidein_25498_139089762@n3303.hyak.local using protocol osdf. Details: Pelican Client Error: failed upload to ap40.uw.osg-htc.org:8443: Request failed (HTTP status 423) (100ms since start) (Version: 7.15.1; Site: UW-IT) (URL file = osdf:///ospool/ap40/data/XXXXXXXX/Leukemia_project/mapped_reads.tar.gz)|

12633237.34371217 XXXX 5/29 11:06 Transfer input files failure at execution point slot1_4@glidein_431023_509537431@node077.cluster using protocol osdf. Details: Error occurred when querying for metadata: Get "https://osg-htc.org/.well-known/pelican-configuration": read tcp 10.1.0.77:60654->104.21.71.171:443: read: connection reset by peer (URL file = osdf:///ospool/ap40/data/ XXXX /unzip)|

12637464.20 XXXX 5/30 07:47 Transfer input files failure at the execution point using protocol osdf. Details: Pelican Client Error: Attempt #3: from dtn-pas.kans.nrp.internet2.edu:8443: request failed (HTTP status 404): server returned 404 Not Found (100ms elapsed, 300ms since start); Attempt #2: from osdf-uw-cache.svc.osg-htc.org:8443: request failed (HTTP status 404): server returned 404 Not Found (0s elapsed, 100ms since start); Attempt #1: from osdf1.chic.nrp.internet2.edu:8443: request failed (HTTP status 404): server returned 404 Not Found (0s since start) (Version: 7.16.5; Site: NotreDame) (URL file = osdf:///ospool/ap40/data/XXXX/freesurfer-v7.2.0.sif)|

So your Pelican-power download failed…

12633237.34981197 XXXXXX 5/30 02:23 Transfer input files failure at the execution point using protocol osdf. Details: Pelican Client Error: Attempt #3: from dtn-pas.kans.nrp.internet2.edu:8443: dial tcp [2001:468:2807::5]:8443: connect: connection refused (10s elapsed, 30s since start); Attempt #2: from dtn-pas.hous.nrp.internet2.edu:8443: dial tcp 163.253.29.19:8443: i/o timeout (10s elapsed, 20s since start); Attempt #1: from osg-houston-stashcache.nrp.internet2.edu:8443: dial tcp 163.253.74.2:8443: i/o timeout (10s since start) (Version: 7.16.5; Site: UChicago) (URL file = osdf:///ospool/ap40/data/ XXXXXX/unzip)|

12633237.35036992 XXXXXX 5/30 03:57 Transfer input files failure at execution point slot1_15@glidein_54145_51861151@n3402.hyak.local using protocol osdf. Details: Pelican Client Error: Attempt #2: from dtn-pas.denv.nrp.internet2.edu:8443: failed to verify size of downloaded file on disk: file size on disk 28671565b does not match expected size 28655181b (2m8.7s elapsed, 4m18.7s since start); Attempt #1: from ncar-cache.nationalresearchplatform.org:8443: Transfer.SlowTransfer Error: Error code 6002: cancelled transfer, too slow; detected speed=27.4 KB/s, total transferred=6.6 MB, total transfer time=2m10.001s (2m10s since start) (Version: 7.16.5; Site: UW-IT) (URL file = osdf:///ospool/ap40/data/XXXXXX/chunkout/nt_virus_chunks/nt_virus_subset_zkwt)|

Mea Culpa…

Yes, we have work to do to improve the error messages…

Yes, we should have better tools to aggregate/filter these messages…

Yes, we are putting structured data in an unstructured string…

Yes, presenting the user with an error message they can resolve is a problem in the first place…

What we want to avoid

- The Pelican team wants to provide enough structured failure information about what happened that you avoid the trap of:

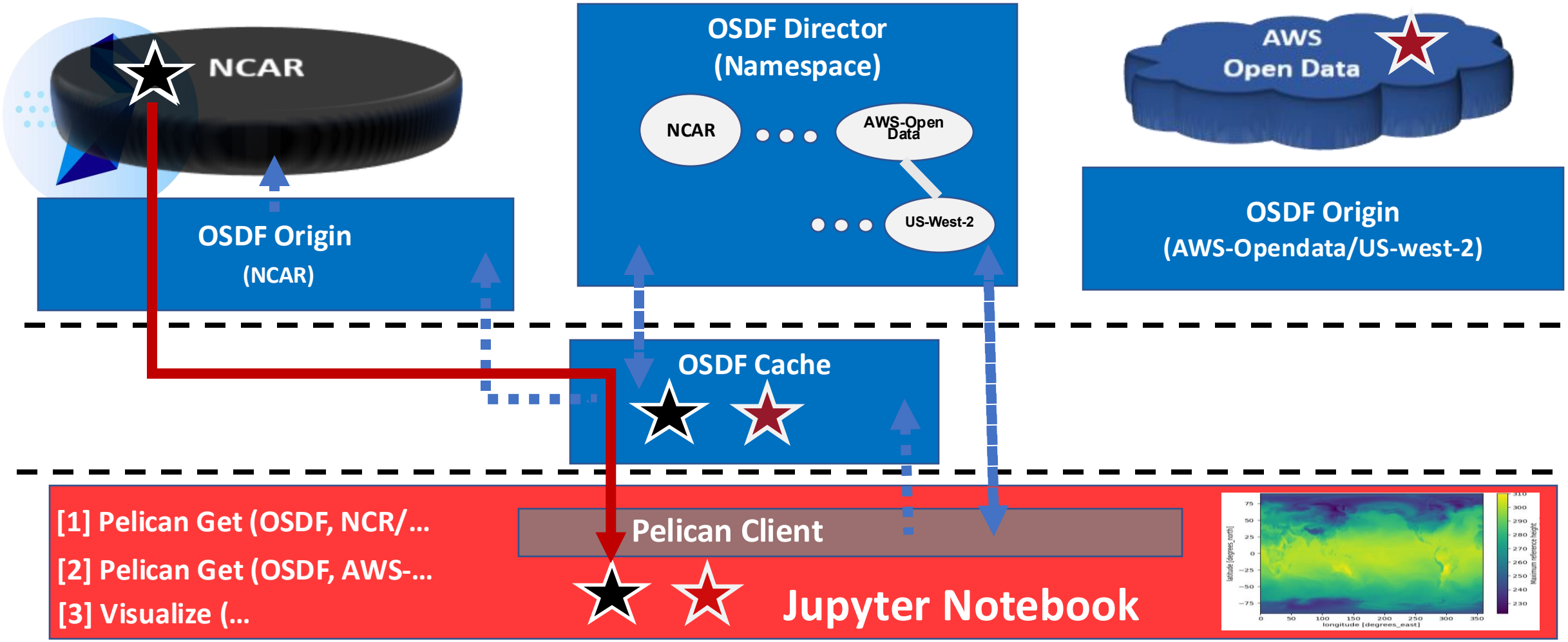
  “Let me rerun is using pelican object copy --debug and reading the tea leaves”

Thinking Logically about Failures

Downloading one byte of data requires 2-3 services to interact successfully.

Understanding the basic architecture is essential for understanding what's gone wrong:

- **Service discovery**: Used to find the director service.
- **Director**: Contacted by the client to discover a service for performing the desired operation.
- **Cache**: Selected by the director, sends the object to the client.
- **Origin**: On cache miss, sends the object to the cache
  - For uploads, contacted directly by the client.



Researcher uses a Jupyter Notebook to create a visualization that requires two objects:

★ NCAR/ rda/harshah/osdf_data/HadCRUT.5.0.2.0.analysis.summary_series.global.monthly.zarr

★ AWS-OpenData/US-West-2/cmip6-pds/CMIP6/CFMIP/NCAR/CESM2/aqua-4xCO2/r1i1p1f1/Amon/co2mass/gn/v20190816

Picking apart a hold message

12633237.35036992 XXXXXX 5/30 03:57 Transfer input files failure at execution point slot1_15@glidein_54145_51861151@**n3402.hyak.local** using protocol osdf. Details: Pelican Client Error: Attempt #2: from dtn-pas.denv.nrp.internet2.edu:8443: failed to verify size of downloaded file on disk: file size on disk 28671565b does not match expected size 28655181b (2m8.7s elapsed, 4m18.7s since start); Attempt #1: from ncar-cache.nationalresearchplatform.org:8443: Transfer.SlowTransfer Error: Error code 6002: cancelled transfer, too slow; detected speed=27.4 KB/s, total transferred=6.6 MB, total transfer time=2m10.001s (2m10s since start) (Version: **7.16.5**; Site: **UW-IT**) (URL file = **osdf:///ospool/ap40/data/XXXXXX/chunkout/nt_virus_chunks/nt_virus_subset_zkwt**)|

Information about the transfer:

- Host: **n3402.hyak.local**
- Site: **UW-IT**
- Pelican Version: **7.15.6**
- URL: **osdf:///ospool/ap40/data/…**

Which do you think is useful?

Picking apart a hold message

12633237.35036992 XXXXXX 5/30 03:57 Transfer input files failure at execution point slot1_15@glidein_54145_51861151@n3402.hyak.local using protocol osdf. Details: Pelican Client Error: **Attempt #2**: from **dtn-pas.denv.nrp.internet2.edu:8443**: failed to verify size of downloaded file on disk: file size on disk 28671565b does not match expected size 28655181b (2m8.7s elapsed, 4m18.7s since start); **Attempt #1**: from **ncar-cache.nationalresearchplatform.org:8443**: Transfer.SlowTransfer Error: Error code 6002: cancelled transfer, too slow; detected speed=27.4 KB/s, total transferred=6.6 MB, total transfer time=2m10.001s (2m10s since start) (Version: 7.16.5; Site: UW-IT) (URL file = osdf:///ospool/ap40/data/XXXXXX/chunkout/nt_virus_chunks/nt_virus_subset_zkwt)|

If the Pelican client considers the error non-fatal, it'll make **3 attempts** to download an object. From above:

Attempt #1:

Service: ncar-cache.nationalresearchplatform.org:8443
Error: Transfer.SlowTransfer Error: Error code 6002: cancelled transfer, too slow; detected speed=27.4 KB/s, total transferred=6.6 MB, total transfer time=2m10.001s.
Timing: 2m10s since start

Attempt #2:

Service: dtn-pas.denv.nrp.internet2.edu:8443
Error: failed to verify size of downloaded file on disk: file size on disk 28671565b does not match expected size 28655181b
Timing: 2m8.7s elapsed, 4m18.7s since start

The second attempt was considered fatal!

Step one: Client finds a service

- The client first discovers the location of the director service from a static file hosted on CloudFlare.

32684753.93699 XXXXXX 5/31 11:08 Transfer output files failure at execution point slot1_7@glidein_48968_35485290@n3353.hyak.local using protocol osdf. Details: **Federation metadata discovery failed** with HTTP status 502. Error message: Cloudflare encountered an error processing this request: Bad Gateway (URL file = osdf:///ospool/ap20/data/XXXXXX/ClusterResult0to50_93405.RData)|

- The client asks the director to select a service (cache) to do the work.

32684753.26096 XXXXXX 5/28 06:31 Transfer input files failure at execution point slot1_11@glidein_1199_243426792@compute38 using protocol osdf. Details: **failed to get namespace information** for remote URL osdf:///ospool/ap20/data/XXXXXX/Result_31513.RData: error while querying the director at https://osdf-director.osg-htc.org: Get "https://osdf-director.osg-htc.org/ospool/ap20/data/ahl/GridGraphs/Result_31513.RData": **dial tcp [2607:f388:2200:c3::3]:443: connect: network is unreachable** (URL file = osdf:///ospool/ap20/data/XXXXXX/Result_31513.RData)|

What needs to happen to send one byte?

For a given cache, what needs to work to send a single byte:

- DNS lookup of the service name.
- Establish TCP connection from client to server.
- TLS handshake.
- Client sends HTTP request to server.
- Server sends HTTP response headers.
- Server sends one byte of data.

How can this go wrong?!?

DNS, TCP, TLS, HTTP

DNS: dial tcp: **lookup** fdp-d3d-cache.nationalresearchplatform.org on 10.24.255.254:53: server misbehaving

TCP: dial tcp [2607:f388:2200:c3::3]:443: **connect**: network is unreachable

TLS: net/http: **TLS** handshake timeout

HTTP: timeout waiting for **HTTP response** (TCP connection successful)

DNS, TCP, TLS, HTTP

DNS: dial tcp: **lookup** fdp-d3d-cache.nationalresearchplatform.org on 10.24.255.254:53: server misbehaving

Text generated by OS

TCP: dial tcp [2607:f388:2200:c3::3]:443: **connect**: network is unreachable

Text generated by Go (programming language) runtime

TLS: net/http: **TLS** handshake timeout

Text generated by Pelican team

HTTP: timeout waiting for **HTTP response** (TCP connection successful)

One byte went through – now what?

- Once a HTTP/1 server sends its response headers, it must send the full body.
  … what happens if there is a read error on byte 2?
- There is no “post-header” error signal. Only option is for the HTTP server to abruptly close the connection: the dreaded EOF (“end of file”) error.

Attempt #2: from dtn-pas.hous.nrp.internet2.edu:8443: unexpected **EOF** (4s elapsed, 14.1s since start)

The following are identical in HTTP/1:

- The origin encountered a read error.
- The cache encountered a read error.
- There was a network connectivity issue at the client.

Do you want to debug the network connectivity at every possible client location?!?

A small tweak on HTTP

- To help differentiate between failure cases, we allow the Pelican client to indicate an error after the download starts (opt-in).
  - Translates to an error message like this:

transfer error: Unable to read (...Path...); timer expired

A tweak on HTTP

- To help differentiate between failure cases, we allow the Pelican client to indicate an error after the download starts (opt-in).
  - Translates to an error message like this:

transfer error: Unable to read (...Path...); timer expired

Writing error messages are hard! In English, this translates to “origin timed out when bytes were requested by the cache”.

Back to the beginning

Is “condor_q –held | grep osdf” good?

- Grep’ing through a bunch of error messages to poke at failures randomly is not particularly structured thinking!

What are some better approaches? Ideas:

- Use ‘condor_history’ to view individual ClassAds.
- Find a friend running ElasticSearch and “condor_adstash”

condor_history knows all!

- The “-transfer-history” flag allows you to pick through all the individual attempts.
- You’re welcome to attempt to be a “command line junkie” to script this output!

bbockelm — bbockelm@ospool-ap2140:~ — ssh ◂ ssh ap40.chtc.wisc.edu — 114×38

```
[bbockelm@ap40 ~]$ condor_history -json -transfer-history -limit 10 | head -n 55
[
{
  "ProcID": 37839140,
  "ClusterID": 12633237,
  "TransferClass": "INPUT",
  "EpochWriteDate": 1748911613,
  "NumShadowStarts": 1,
  "MachineAttrName0": "slot1@glidein_2_91164458@e4039.chtc.wisc.edu",
  "InputPluginResultList": [
    {
      "TransferUrl": "osdf:///ospool/ap40/data/daniel.morales/unzip",
      "TransferType": "download",
      "DeveloperData": {
        "Attempts": 1,
        "DataAge0": 1534602.0,
        "Endpoint0": "osdf1.chic.nrp.internet2.edu:8443",
        "TransferTime0": 0.055,
        "ServerVersion0": "XrootD/v5.8.2",
        "ClientChecksums": {
          "crc32c": "40830b29"
        },
        "TimeToFirstByte0": 0.0,
        "TransferEndTime0": 1748911610,
        "TransferFileBytes0": 512668,
        "PelicanClientVersion": "7.16.5"
      },
      "TransferEndTime": 1748911611,
      "TransferSuccess": true,
      "TransferFileName": "unzip",
      "TransferProtocol": "osdf",
      "TransferFileBytes": 512668,
      "TransferStartTime": 1748911610,
      "TransferTotalBytes": 512668
    },
    {
      "TransferUrl": "osdf:///ospool/ap40/data/daniel.morales/piRNAs/vpiRNA_subset_0202",
      "TransferType": "download",
```

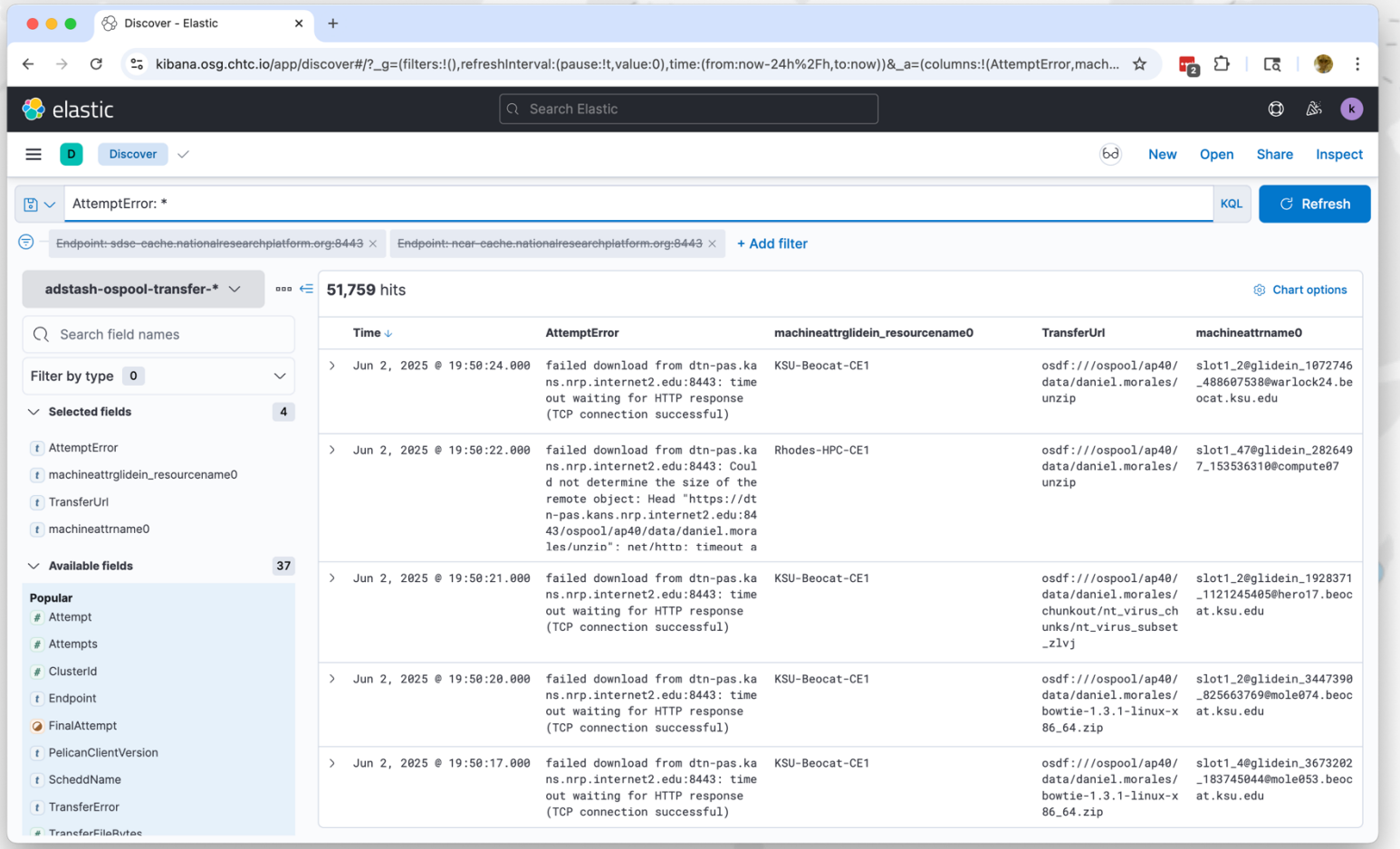
condor_history knows all!

- The “-transfer-history” flag allows you to pick through all the individual attempts.
- You’re welcome to attempt to be a “command line junkie” to script this output!
  - One default output method is JSON which is particularly scriptable.

bbockelm — bbockelm@ospool-ap2140:~ — ssh ◂ ssh ap40.chtc.wisc.edu — 114×38

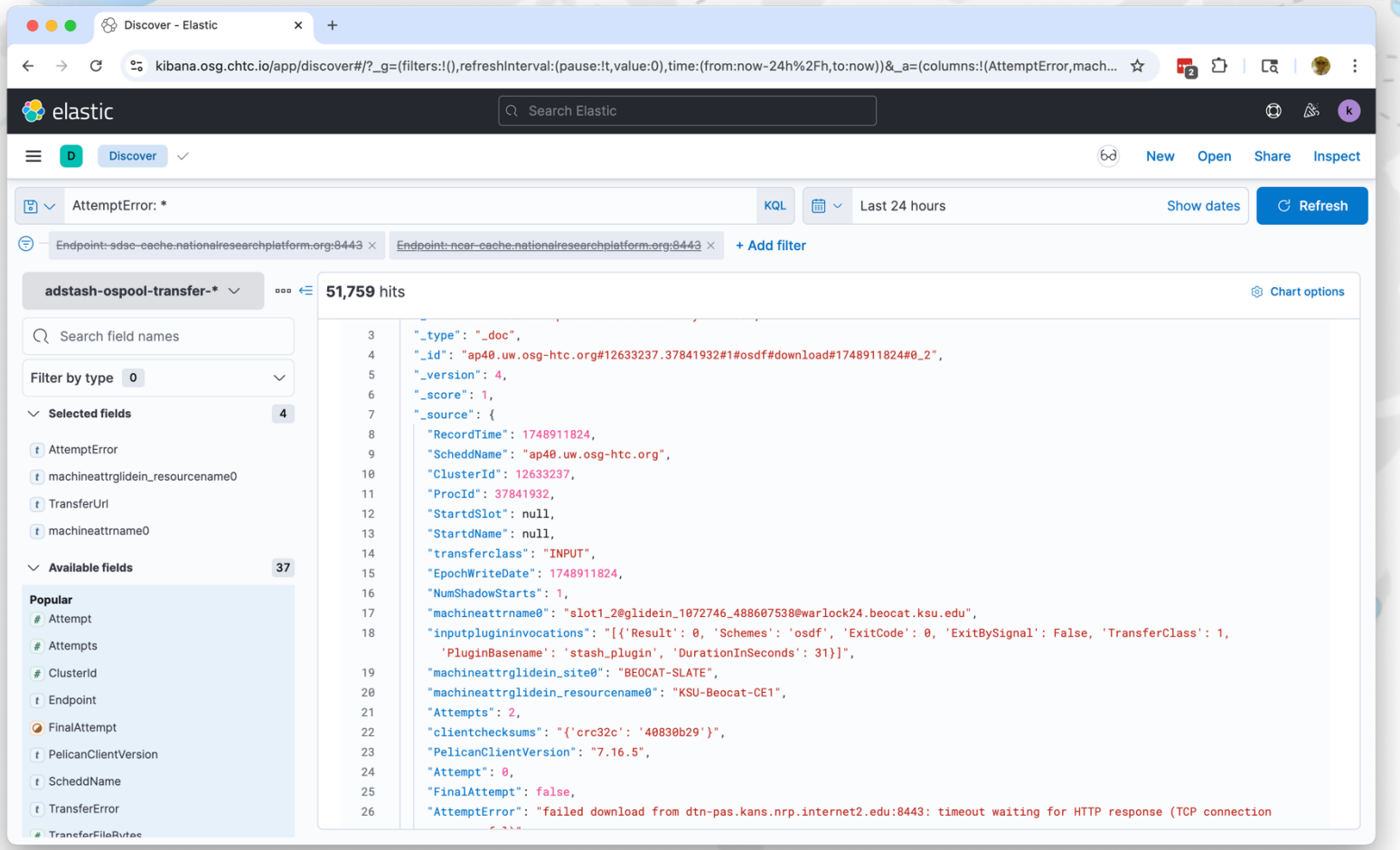
```
[bbockelm@ap40 ~]$ condor_history -json -transfer-history -limit 10000 | jq '.[] | .InputPluginResultList | .[] | ]
select( .TransferSuccess == false ) '
{
  "TransferUrl": "osdf:///ospool/ap40/data/giorgia.nicolaou/my-inference-container/inference-v2.sif",
  "TransferType": "download",
  "TransferError": "failed to get namespace information for remote URL osdf:///ospool/ap40/data/giorgia.nicolaou/m
y-inference-container/inference-v2.sif: error while querying the director at https://osdf-director.osg-htc.org: Ge
t \"https://osdf-director.osg-htc.org/ospool/ap40/data/giorgia.nicolaou/my-inference-container/inference-v2.sif\":
 dial tcp: lookup osdf-director.osg-htc.org: i/o timeout",
  "TransferSuccess": false,
  "TransferFileName": "inference-v2.sif"
}
{
  "TransferUrl": "osdf:///ospool/ap40/data/daniel.morales/unzip",
  "TransferType": "download",
  "TransferError": "failed to get namespace information for remote URL osdf:///ospool/ap40/data/daniel.morales/unz
ip: error while querying the director at https://osdf-director.osg-htc.org: Get \"https://osdf-director.osg-htc.or
g/ospool/ap40/data/daniel.morales/unzip\": dial tcp 128.105.82.132:443: i/o timeout",
  "TransferSuccess": false,
  "TransferFileName": "unzip"
}
{
  "TransferUrl": "osdf:///ospool/ap40/data/giorgia.nicolaou/my-inference-container/inference-v2.sif",
  "TransferType": "download",
  "TransferError": "failed to get namespace information for remote URL osdf:///ospool/ap40/data/giorgia.nicolaou/m
y-inference-container/inference-v2.sif: error while querying the director at https://osdf-director.osg-htc.org: Ge
t \"https://osdf-director.osg-htc.org/ospool/ap40/data/giorgia.nicolaou/my-inference-container/inference-v2.sif\":
 dial tcp: lookup osdf-director.osg-htc.org: i/o timeout",
  "TransferSuccess": false,
  "TransferFileName": "inference-v2.sif"
}
{
  "TransferUrl": "osdf:///ospool/ap40/data/giorgia.nicolaou/my-inference-container/inference-v2.sif",
  "TransferType": "download",
  "TransferError": "failed to get namespace information for remote URL osdf:///ospool/ap40/data/giorgia.nicolaou/m
y-inference-container/inference-v2.sif: error while querying the director at https://osdf-director.osg-htc.org: Ge
t \"https://osdf-director.osg-htc.org/ospool/ap40/data/giorgia.nicolaou/my-inference-container/inference-v2.sif\":
 dial tcp: lookup osdf-director.osg-htc.org: i/o timeout",
```

Not everyone loves the CLI, Brian!



- More of a database person?
- ElasticSearch provides a document-centric data model from a browser environment.

Not everyone loves the CLI, Brian!



- More of a database person?
- ElasticSearch provides a document-centric data model from a browser environment.

So what did we learn?

Some thoughts…

1. Don’t bother memorizing error messages. We are trying to constantly change and improve them.
    1. Instead: complain to us about how we can communicate better!
2. Think through the logical steps of what Pelican is doing. Where, precisely, did the error happen?
    1. Director versus a cache?
    2. DNS, TCP, TLS, or HTTP?
    3. Before bytes moved or after?
    4. Which can you control versus just retry?
    5. The Pelican signals to HTCondor when it believes the error is retryable.
3. Let HTCondor run the transfer so you can state your retry policy.
4. Spend more time thinking about aggregate errors and less about the individual failures.
5. Reach out to the Pelican team with ideas for better tools!

Questions?

This project is supported by the National Science Foundation under Cooperative Agreements OAC-2331480. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.