



dCache news

Dmitry Litvintsev

FIFE meeting

05/26/23

dCache update on 05/17/23

- We applied a patch that handles xrootd access so that:
 - Access with a token can use “(x)root” schema (without “s” at the end) on default port (1094)
 - This allows to access data using token from root analysis framework
 - Verified by mu2e to be working properly (kudos to Ray Culbertson)
 - Additionally access by older xrootd-client (< 5) is handled uniformly on the default port.

Issues with listing

- In the recent weeks we are observing rather heavy listing activity that is usually caused by PROPFIND http method (but not exclusively, we see GFTP listings running too) from batch jobs.
- IMO there should be no reason to run listings from batch jobs under any circumstances.
- Example, from dCache internal monitor:

```
duneana 2023-05-15 14:46:08.782 (116 s ago) 2023-05-15 14:46:08.782  
(116 s ago) /pnfs/fnal.gov/usr/dune/scratch/users/erea/grid_jobs
```

```
ls /pnfs/fnal.gov/usr/dune/scratch/users/erea/grid_jobs | wc -l  
124646
```

LIST/PROPFIND calls

- I have been watching the ls activity Here is an example.
This command:

```
ENC{LIST  
/pnfs/fnal.gov/usr/nova/scratch/users/rkralik/TBCalib/At  
tenProf_Period2Data}
```

is being executed with average frequency of 12 per minute.

This directory has about 5K entries. This is just one GFTP user.

- frequency of LIST commands by GFTP - is 5 Hz
- frequency if PROPFIND calls by WebDAV is 88 Hz

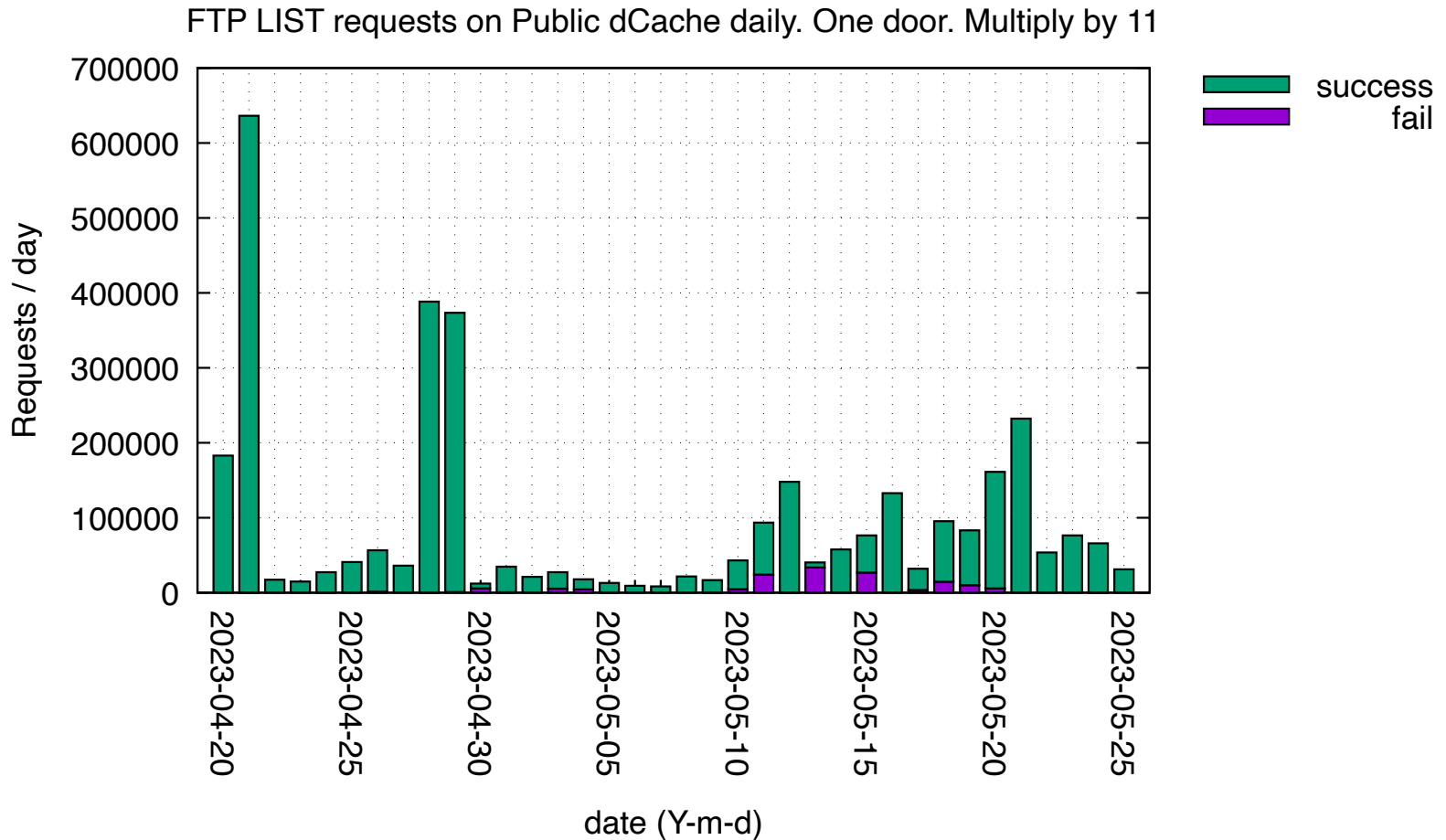
Bad practice

- We are seeing that there are cases where destination path (for instance) is checked for existence by running “ls” on it:

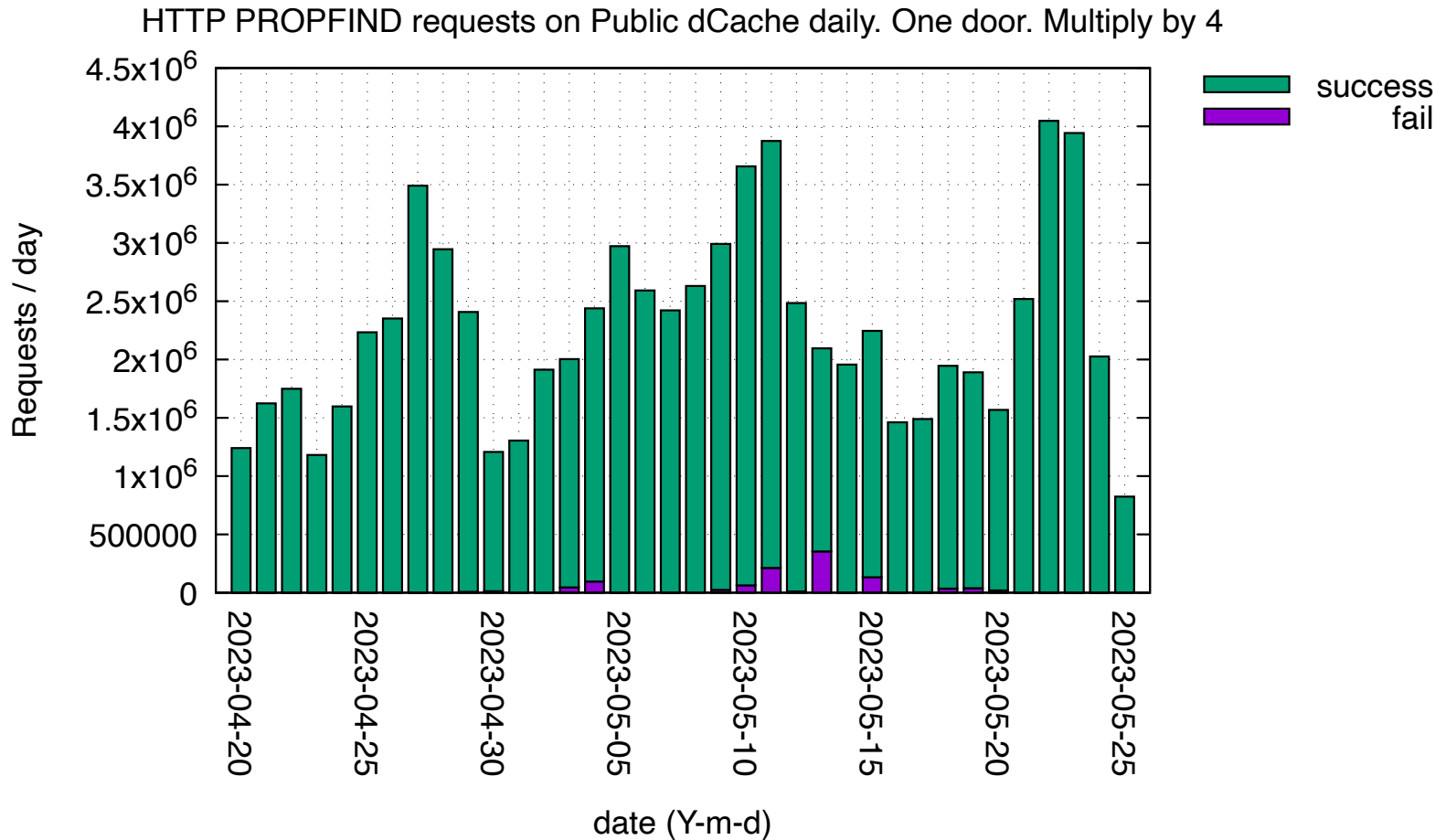
```
print("runNovaSAM is making a directory with IFDH")
dh = ifdh.ifdh("http://samweb.fnal.gov:8480/sam/nova/api")
print("Checking if directory ", dir, " exists")
files = dh.ls(dir, 1)
```

- This is bad idea! And it gets worse when it is done from thousands of jobs simultaneously.
- dCache supports autocreation of destination, so no checks like these are even necessary.

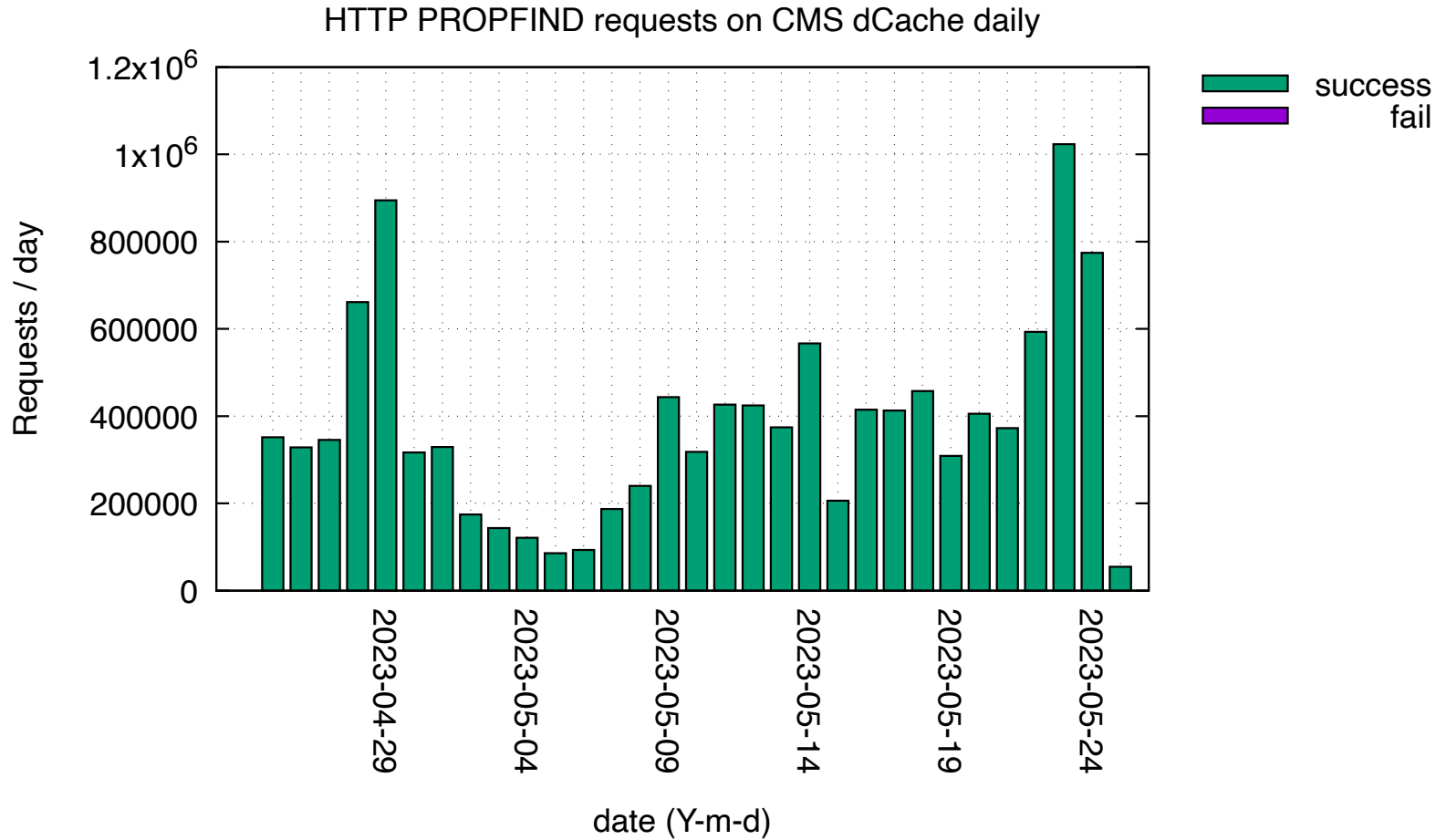
FTP List request daily on one door. Multiply by 11



HTTP PROPFIND calls on Public dCache. Multiply by 4



HTTP PROPFIND calls on CMS T1 Disk. (for comparison)



What have we done on dCache end

- We have revamped handling of list requests in dCache
 - Before:
 - N threads would process single list queue
 - PROBLEM: N long running list requests even in the same directory would starve N threads.... **Everybody stuck**
 - Now (since 05/21/23):
 - N threads process N list queues. Each one thread handles one and the same queue.
 - Repeated requests to list a path are dispatched to the same queue. So, simultaneous lists of the same directory are serialized.
 - Once listing of a path has completed, the queue is checked for requests for the same path and user. All these requests are collapsed using the result of just completed listing
 - RESULT:
 - Significant improvement in LIST request processing. **“LS on steroids”** really.
 - Long running lists do not starve threads.
 - A downside – within duration of the single listing, all arrived list requests will get the same data. We consider this to be a non-issue.

Conclusion

- In the last few weeks we have seen degradation of service caused by excessive listings of large directories executed on massive scale by batch jobs.
 - It is not known if this just a random spike of large directory listings or more massive use of HTTP (so “something in gfal”). But we occasionally seen this in the past as well.
 - The list handling has not changed in years on dCache end.
- We communicated with IFDH support and individual users and identified cases where listings are used just to check directories existence. Some of these checks are done recursively (walking down the full directory path).
 - Recommendations have been made.
- At the same time we significantly improved handling of LIST requests in dCache.