ENDIT news

NeIC NT1 Manager
Mattias Wadenstein <maswan@ndgf.org>
Niklas Edmundsson <nikke@hpc2n.umu.se>

2026-05-07

NORDIC E-INFRASTRUCTURE COLLABORATION

Overview

- History
- Performance enhancements
- Bugfixes
- Future work

Where did we come from?

Origins

- April 2006, NDGF-T1 needed tape for DC4
  - Most of our sites also run TSM for backups
  - TSM only does request reordering within a session (i.e. one invocation of dsmc archive/retrieve)
  - We created a hsmscript and daemons that used intermediary directories for batching requests
  - These directories were the API between the hsm script and background processing, still much the same today
- Contributors:
  - Mattias Wadenstein
  - Lars Viklund
  - Niklas Edmundsson

Plugin

- Running a shell script per request had scalability issues
  - Our typical deployment is one read and one write pool per library and experiment
  - Remember: Two hosts can’t coordinate the request reordering in TSM
- Running out of file descriptors in the pool, lots of memory and CPU for >10k shell scripts polling, etc
- Gerd Behrmann wrote a plugin
  - And created a dCache hsm plugin API to go with the plugin :)
  - Uses the same filesystem API as the hsm script
  - Java threads and/or async instead of shell processes
  - Initial commit 2014-06-28
- Contributors:
  - Gerd Behrmann, Vincent Garonne, Krishnaveni Chitrapu, Niklas Edmundsson, Tigran Mkrtchyan

Scheduling improvements

- We needed to use more tape drives to keep up
- So we needed to split requests by tape for reads or by size for writes
  - Instead of throwing all requests into one dsmc invocation
  - Required to know what files are on what tape
  - Luckily dsmc can query this, and we create a daily hintsfile
  - Files not in the hintsfile are treated as being on the tape “default”
  - 2016-04-08, initial commit for tapehints generator

Other improvements

- Remountdelay to not overwork tapes
  - If you request a file on a tape that’s been mounted recently: wait
- Wait a while during incoming requests before sending a batch
- Getting IBM to fix a dsmc memory management bug
  - Took ~O(n³) by number of requests, noticeable from around 10k, previously reported in this forum
- Batching deletes
- “Performance improvements and bugfixes”

2.0 at 20 years old!

Where are we going?

2.0!

- Current development goals:
- Make sure we have a decent amount of work before we mount a tape
- Minimum performance targets:
  - On a small dedicated pool with 1TB reasonably fast SSD:
  - At least 100k outstanding requests on a single pool
  - At least 100Hz rate per pool
  - Maintaining full tape drive speed
  - No unnecessary tape remounts due to ENDIT limitations

Goals: Our intended layout

- Designed for one tape read and write pool per experiment and tape library
  - Multiple write pools work fine, if needed
  - Multiple read pools doesn’t work good with TSM
    - Reordering is per session so you’d just end up with lower efficiency and readers waiting for tapes that the other pool is accessing
    -
- For optimal performance: One fast SSD node per pool
  - Important to never be so slow that the tape drives have to break

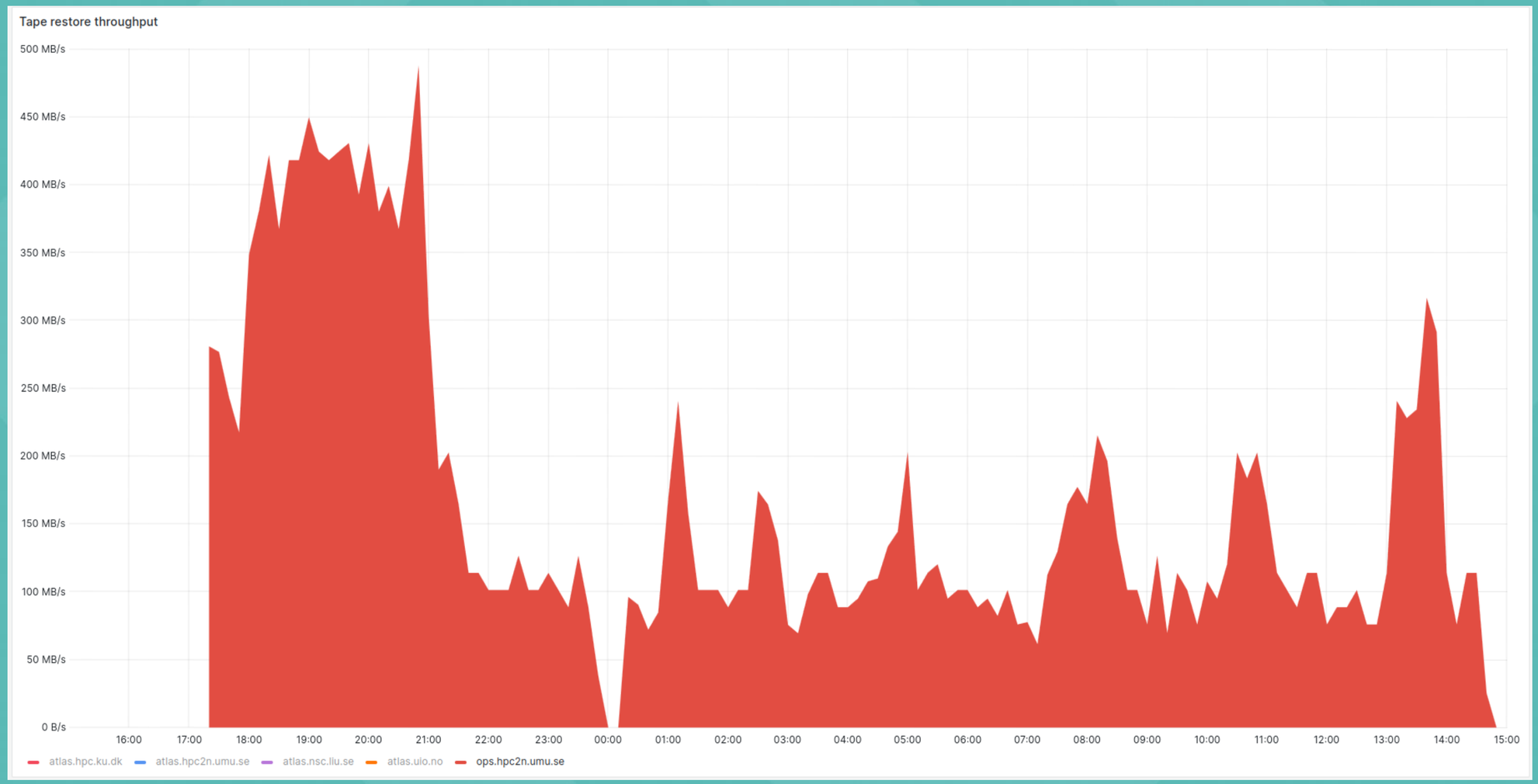
Performance enhancements

- Delay space allocation.
  - Previously space allocation was made when the request was queued, this effectively limited the request queue to the pool size.
  - Given the trend with smaller flash-based tape pools and bigger tapes this resulted in lots of remounts and ineffective tape use.
  - Space is now allocated when the provider sees a non-zero sized file, i.e. when staging of a file has begun.
  - 2023-12-06 proof of concept in place

Latealloc

- Glossary
  - Activate: Tell dCache the request is active and create the request file
  - Start: The file in “in/” exists
  - Allocate: Reserve space in the pool
  - Complete: File is of correct size (and we have waited 1s for dsmc to finish up), the file gets renamed into the pool directory
  - Checksum: The plugin ends with a successful restore done, and returns an empty set of checksums from the tape library (not supported by TSM)

Latealloc benchmark, 10%



Bugfixes

- Eliminating the 1-Hz-per-thread issue.
  - Root cause was a subtle but obvious thing that has gone unnoticed by multiple reviewers: Doing Thread.sleep releases the thread on an OS level but still locks execution in the application.
  - The sleep is needed to allow for dsmc setting file permissions on the staged file before completing staging by moving it to the dCache pool directory.
  - Now reschedules the thread instead.
  - Also the sleep duration is now configurable, including disabling it for non-dsmc/ENDIT daemons environments.

Performance enhancements

- Fix the watching provider to work with real load.
  - Root causes were both the 1-Hz-per-thread issue (previous slide) and the actual behavior of MODIFY events.
  - MODIFY events actually reports an event for each write operation to a file in the watched directory, this essentially produces an event storm at the IO rates we need in production.
  - OS (Linux) inotify has IN_CLOSE_WRITE that would be a better match, but the Java event implementation doesn't implement it.
  - Now only uses CREATE events to detect staging starts and polling to detect staging completion.

Performance enhancements

- Benchmarked at more than 250 Hz staging rate.
  - Modest thread counts, new defaults are good for production.
  - Benchmark used small files, but complete workflow from TSM-server to pool.
- Plugin build now includes version info.
  - Git properties file with describe info.
  - Use “hsm show providers” in dCache admin interface to display it.
- Plugin has debug info
  - Set appropriate logging levels to get verbose stuff

Future work

- More testing, rolling out on production pools.
  - Get a grip of a weird issue with canceled requests (we think it's one plugin bug and one dCache bug).
  - More testing, rolling out on production pools at NDGF
- Public beta
- Stable release
- Input from people running other backends with the plugin
  - Add these to the readme and merge local plugin customizations
- Better handling of files spanning multiple tapes

Questions?

NORDIC E-INFRASTRUCTURE COLLABORATION