Reclaim CephFS Capacity Through Pool Tiering + Migration

Ceph Day Seattle 2026
2026

Anthony D'Atri, Master of Tentacles
Ceph Ambassador + Documentation Lead

Who's Anthony?

- Ceph at scale since 2014
- (Successfully since 2017)
- Ceph Ambassador
- Ceph Documentation Lead
- Author: *Learning Ceph, Second Edition*
- QLC fanboy
- Ceph is quite graphically a part of me

Data expands to fill available storage (and beyond)!

- It used to be that enterprise storage meant 6U Fujitsu 2351 Eagles, holding a mind-boggling 380 MiB of data: enough for a whole company!
- Today that 380 MiB can’t store a 4k pickleball video
- Enterprises, educational institutions, and really just about anyone these days demand storage capacities that **start** on the order of hundreds of tebibytes and rapidly grow to pebibytes.

Data expands to fill available storage (and beyond)!

- As this article is written in the spring of 2026, the memory market, which includes DRAM, SSDs, and legacy HDDs, has experienced a dramatic escalation of pricing.
- It is not uncommon to be quoted a price four times what the same hardware cost a year ago, and there are signs that it is going to get worse before it gets better.

What’s a poor ammonite to do?

- Thus, many Cephers find themselves between the Charybdis of quotes calculated via Disaster Area's hypermathematics and the Scylla of hungry users armed with torches and git forks.
- Short of nuking the site from orbit, how do we make everyone happy, or at worst mildly discontented?
- Efficiency!

CephFS

- CephFS is a popular, highly available and scalable software defined POSIX-style distributed filesystem that manages tens of pebibytes of precious data
- Ceph deployments often begin small, with replication for perceived performance needs
- As the cluster grows to more nodes and more data, it may become desirable to switch to Erasure Coding (EC) to make more efficient use of raw capacity

CephFS with erasure coding

- This Erasure Coding overhead table presents efficiency (space amplification) factors for a spectrum of EC profiles https://tinyurl.com/mpfs8jvc
- Replicated pools usually maintain three copies of data: overhead (space amplification) factor of 3.0
- EC 4+2 or 6+3 for all but the tiniest files yields a space amp factor of just 1.5, with some tradeoffs.
- An EC pool thus can require substantially less raw storage for a given amount of user data, or store gobs more user data on a given amount of raw capacity

More on erasure coding

- EC has traditionally been limited to workloads with modest write performance requirements
- With SSDs, however, many workloads are actually compatible with the tradeoffs of EC
- Especially with the advent of Fast EC in the Tentacle release
- HDDs are often a false economy
  - IOPS / $
  - IOPS / TiB
  - IOPS / RU
  - PiB / RU
  - Recovery time (time = risk!)

More on erasure coding

- Cephers often think that once data is written to a RADOS pool, the data protection strategy is carved in stone unless one manually and disruptively moves data to a different pool
- RGW Lifecycle policies offer a way to transition object data between storage classes and pools transparently, but we're to talk about the draft ^H^H^H erm … CephFS
- Cephfs volumes provision one RADOS pool for metadata (foo.meta) and at least one for data (foo.data)

EC efficiency

- Consider a community member storing breathtaking amounts of data, adding more every day
- All stored on mainstream TLC SSDs in Replica 3 pools, because these are the defaults for performance.
- Quotes for capacity expansion came in **dramatically** higher than budgeted, and they were in a bind.
- While their MDS estate is very busy, at the RADOS level their datalake is read-**mostly**, with modest client traffic
- They need reads to be fast, but writes are of lesser concern.

CephFS erasure coding

- One can attach additional data pools to a CephFS volume, which may live on different media (HDD, mainstream NVMe SSD, coarse-IU QLC SSD) and/or employ different data protection strategies
- If data pools are added to a volume, inode backtraces and HEAD objects are still placed in the first data pool.
- One of the more obscure but very, very useful features of CephFS is the ability to set *file layouts*
- File / dir layouts designate to which data pool new files should be written, with inheritance and overriding
- But what about existing files in existing pools?

Efficiency

- The ultimate answer to EC, CephFS, and everything: a transcoder script posted by Reddit user `marcan42`
- This script is inspirational, with some shortcomings
- Walk a CephFS volume or subdirectory, processing files by creating a copy with the new file layout then atomically renaming the copy over the original
- Clients access data without risk of disruption, corruption, or divergence.
- By working one file at a time, we avoid the need for vast amounts of scratch space

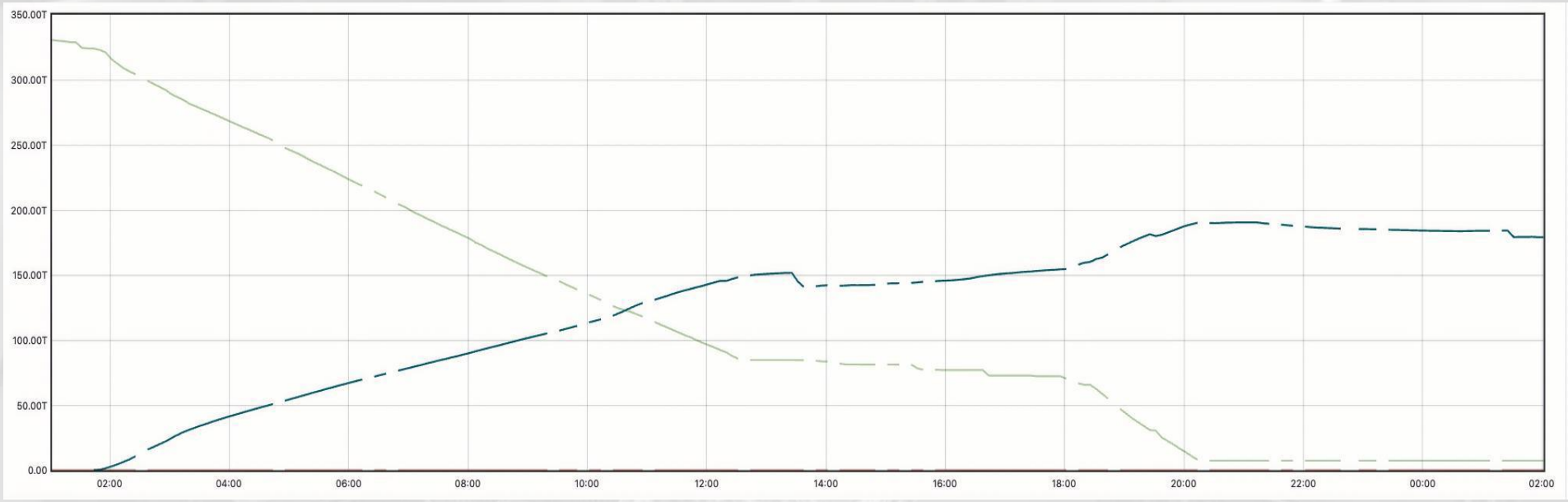
Tiering and migration

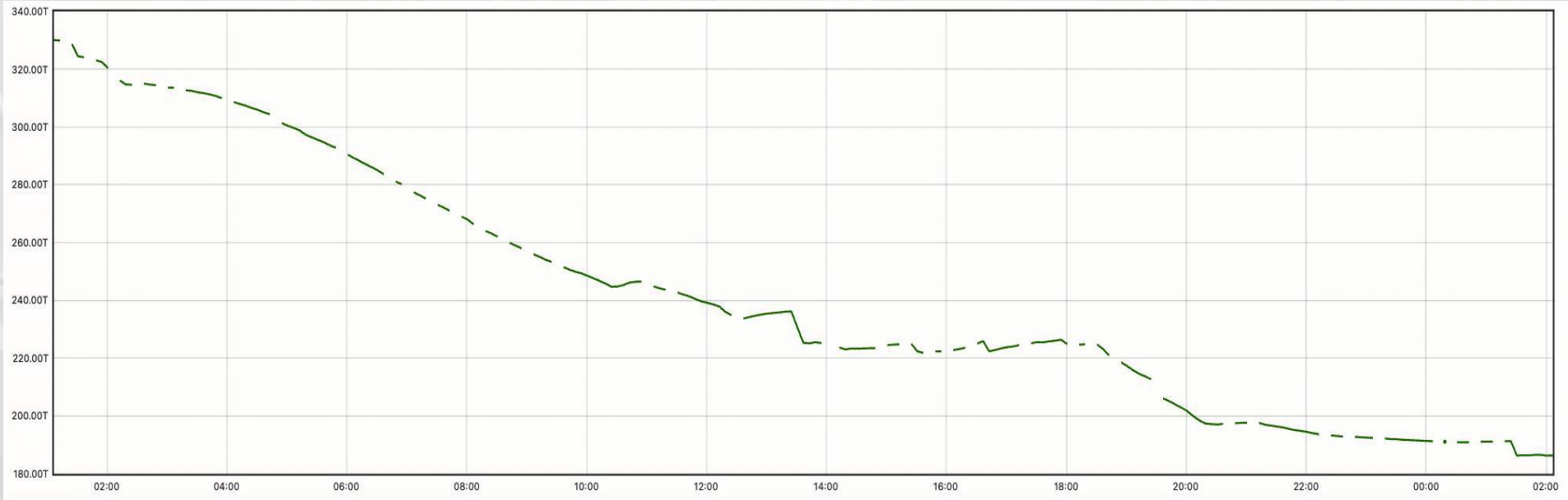
- If we had that kinda free space, we wouldn't be having this conversation
- Painstaking care is taken to avoid divergence or corruption
- One can limit operation to
  - A directory subtree
  - Avoid multiply linked files (abundance of caution)
  - Files older / newer than a given number of days
- By default process files in parallel; adjust to trade off network and client impact against elapsed time
- When using EC, the very smallest files can consume **more** raw capacity than replication.

Real world data

- This fantastic resource from Mark Nelson quantifies and visualizes this space amplification for small files
- It is said that a picture is worth 1024 words, so keep reading
- The next slide shows the used by the original replica 3 data pool in green, and that taken by the new EC 6+3 pool in blue as transcoding progresses

Capacity Reclaimed!





Migration details

- This run specified:
  - 15 threads
  - Minimum 500 KiB file size
  - Minimum 1 day file age
  - Skip multiply-linked files
  - A 6+3 EC policy
- Files were processed at 12-15 TiB per hour
- The script is idempotent
- No need to re-run to pick up newly-created files as the layout for the directory is set to the new data pool
- To-do: pg_num mangement

Fine print

- This process manages EC profiles and CRUSH rules, and creates a new RADOS pool for each CephFS volume transcoded, so be sure to attend to your PG strategy
- To-do: filter on file/dir name regexes
- This has not been tested on CephFS subvolumes, nor has it been tested transcoding back to a replicated volume
- Do not taunt script or stare into it with remaining eye
- Suggestions or improvements are welcome, on a best-effort basis

Executive Summary

- CephFS transcoding enables non-disruptive migration from replicated pools to erasure-coded pools
- Replica 3 to EC 6+3 reduced raw storage consumption by approximately 44%
- The transcoder safely handles active files, symlinks, and hard links
- FastEC improvements in Ceph Tentacle significantly improve EC performance and space amp for tiny files
- This approach reduces hardware spend while extending cluster runway
- A companion script sets up the CephFS volume for tiering. Migration is separate for sanity checking and to allow running on different systems.

Resources

- Script repository: https://github.com/anthonyeleven/vcephfs_transcoder
- Original basis script: https://www.reddit.com/r/ceph/comments/1l1ey87/cephfs_layoutpool_migration_script/
- Author: anthony.datri@gmail.com

Questions?