Testing Starfish at LLNL

Olaf Faaland

September 25, 2018

LLNL-PRES-758168

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

The Current Landscape

- Many of our users need help managing their files
  - They produce many files and directories
    - … For many compute jobs, for potentially many projects
    - … Using many applications and libraries
    - … Over long periods of time
    - … On different file systems, some of which were retired
    - … Sometimes inheriting file collections from others
  - They often know tools that scale poorly
    - ls, find, du, rm
  - They have trouble
    - Finding file sets for specific past compute jobs
    - Knowing where the old, obsolete, sets of files are in their trees
    - Knowing which sets of files are pushing them over quota or filling up the FS

Hopes and Dreams

- We want to provide better visibility to users (and ourselves!)
  - Easily see which directories contain most of their files and directories
  - Easily see the storage needs of jobs, to estimate future usage
  - Find datasets from a given job
    - Via searching existing metadata such as names and dates, like a faster “find”
    - Via marking a file or directory somehow
    - Even when they’ve moved from one file system to another
- We want to automate tasks to make management easier
  - Allow users to mark a directory tree for deletion
    - We can choose an appropriate tool, e.g. MPITools “drm”
  - Allow users to mark a directory tree for transfer elsewhere
    - Tape, another Lustre FS, etc.
  - Generally, we need a signaling mechanism, and the information about the directory tree to enable us to use the right tool in the right way

Our Test Environment

- Test File System “lquake”
  - Lustre 2.8 at first, later 2.10
  - 16 SSD-based MDTs
  - 4 HDD-based OSTs
  - About 745 Million inodes; directories with thousands of files
- Starfish / Lustre Client Node
  - NVME based at first (about 1.5 TB)
  - Filled up NVME, and switched to SATA drives because they were on hand and not spoken for. Unsurprisingly, this did not perform as well!

Speed Bumps

- Teething Pains
  - The changelog monitoring tool had been written to run on the MDS, which does not conform with our security posture
  - Some confusion about how Lustre handles multiple changelog users
- Lustre Issues
  - lfs utility incorrectly parsed hex in MDT name
  - lfs changelog does not provide a way to query the current index or the user's index
  - lfs XXX YYY may not return every record between XXX and YYY.
    - At least as of 2.10
    - Records are not stored in order in the LLOG on the MDT, and this seems to be handled incorrectly by the server side request handler

… And More Speed Bumps

- Other obstacles
  - Starfish bugs
  - Re-installed after we filled up the NVME, so needed to re-scan the FS

Results so far

- Scan and Changelog Ingest
  - During our last test, Starfish's scan appeared to be able to finish in about 2 days, but failed when the NVME storage filled up.
  - Starfish read about 500 million changelog entries in under 3 hours (queued for processing, not including time to readdir() or stat() as necessary)
  - Real performance numbers will come in the future.
- Web UI / Reporting
  - The UI looked clear and the results were quick enough for interactive use during casual testing.
  - In particular, browsing through a directory tree seemed viable.
- Tagging Directories
  - The tags are really associated with paths in the Starfish DB; so when a file or directory is moved, the tag is no long associated with it.
  - We believe this may still be useful to our users, though.

Next Up

- Bought new nodes to Starfish specs
- Test with a Lustre 2.10 production file system
  - We are transferring data from an existing Lustre 2.5 file system to a 2.10 file system
  - We will set up Starfish to monitor the new file system
    - Test real performance
    - Get staff using it to help answer user questions
    - Get some end-users to try it out
- Monitor other existing file systems
- Experiment with automating alerts or triggering actions based on Starfish data

Acknowledgements

- The work installing Starfish and testing it was done by Trent D'Hooge (LLNL). I just helped occasionally.
- Starfish developers and support staff were highly available to troubleshoot the various problems we encountered

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.