Fast Lustre File System Scanning

RUG 2017

Cory Spitz

October 3, 2017

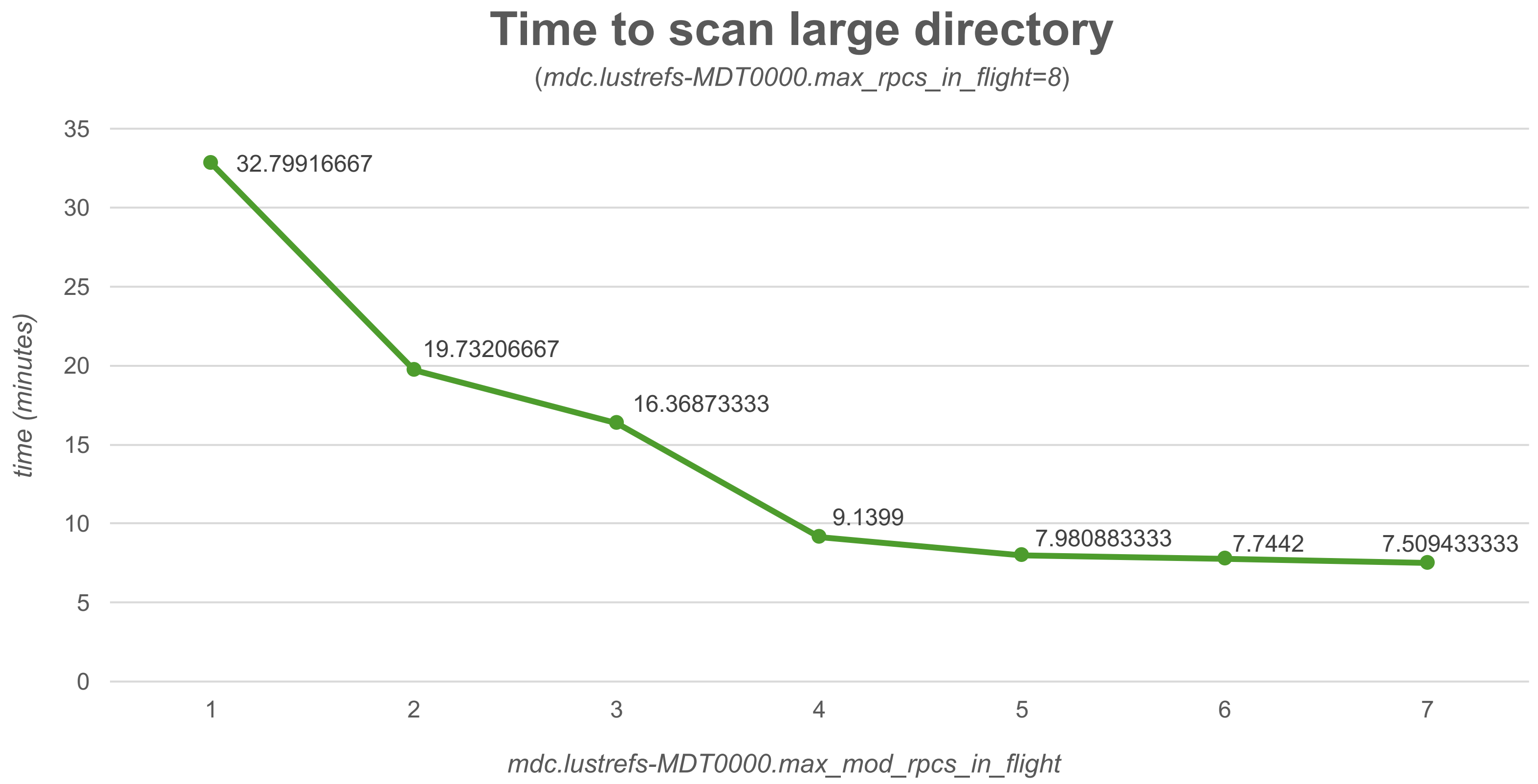
Abstract

It can take Robinhood weeks or months to fully scan a large-scale Lustre file system. Performance could be much better, but the scan rate can be throttled by Lustre clients. Despite the fact that Robinhood doesn’t modify filesystems, scans perform better with the advent of the so-called ‘multiple modifying metadata RPCs in flight’ feature. This is because FID lookup causes open/close activity that was formerly serialized. Even re-mounting the filesystem read-only does not allow Robinhood to bypass this throttling. Cray is working to develop a bypass that will unthrottle client activity for dedicated Robinhood scanning. This talk details our efforts.

Problem: Lustre serialization

- **Typically, an initial scan or rescan is done from a single client**
  - Naturally, scan time increases with file system size
  - Not a snapshot; process can last weeks on large file systems
  - So, we desire better single client performance
- **Performance doesn't increase with larger values of max_rpcs_in_flight**
  - FID lookup via llapi_path2fid() requires an open() to issue an ioctl()
  - That makes it a 'modifying' RPC
- **Even though RH doesn't modify the filesystem, we need max_mod_rpcs_in_flight, available in 2.8.0**

Scaling with multiple modify RPCs per client



Workarounds without multi-mod RPCs?

- **Problem: few servers are at this version**
- **OK, so re-mount read-only**
  - That doesn't work!
  - Need to protect the potential recovery of open-unlinked files
- **What about alternate lookup methods?**
  - We looked at re-implementing path2fid with fid_from_lma()
  - No faster, still needed to get at the xattr
  - FID lookup is just inefficient on clients, (FID-in-dirent is not exposed to clients)
- **What about multiple mounts?**
  - Can get us past the MDC semaphore
  - Hard to set up in practice

Scan-only mode

- **Back to read-only?**
  - What if we don't protect recovery of open-unlinked files?
  - Unlink can race with quick open/close
  - Do we care? We might leak orphaned files is all
- **Scan-only mode patch bypasses MDC semaphore**
  - Read-only opens and subsequent closes
  - getxattr too
- **Results in same ~5x speedup as multi-mod RPCs**
- **"Dangerous" patch hasn't been submitted, but available upon request**

But one/few at a time is not the way to go

- **If I had a billion dollars… and wanted to give it away...☺**
  - ...and you asked for one or even ten dollars at a time
  - That operation fundamentally doesn’t scale
  - Similarly, RH is sucking up all this metadata one request at a time
- **We could pack more requests into RPCs**
  - That’s just weak scaling
  - We’re still putting an excessive processing load on the file system
- **There has to be a better way**

Do server side scanning

- **Similar to lfsck, Lester, or Zester…**
  - Iterate through inodes on MDT in an efficient order
  - Leverage FID-in-dirent
- **How to batch updates to consumers?**
  - Partially explored at LAD '16 Dev Summit
  - Sergey Cheremencev has proposed a batch method generating Lustre ChangeLogs
- **See: "HSM Initial Scan Optimization", LAD '17, Day 2**
  - Very promising
  - Could be combined with Size-on-Metadata

Merci!