Enhanced Adaptive Compression in Lustre Status Report

Anna Fuchs, Michael Kuhn

anna.fuchs@informatik.uni-hamburg.de

Research Group Scientific Computing Department of Informatics, Universität Hamburg

07 December 2017



UΗ



Code submission

- Client-server version
 - Compression on the client
 - Decompression on the server
 - Write path
- Submission in Gerrit
- Ticket LU-10026 in Jira, description and TODOs
- Discussion via mailinglist (?)
- Project description at lustre.org

Client-server



CLIENT-SIDE COMPRESSION IN LUSTRE

Client data



Initial page array, pga[page_count]

Chunk stripe, chunksize = ZFS recod size

Get contiguous buffer[chunksize] from cmp_pool

Get many contiguous buffers chunks x buffer[chunksize] from cmp_pool

CLIENT-SIDE COMPRESSION IN LUSTRE

Client data



CLIENT-SIDE COMPRESSION IN LUSTRE

Client data















SERVER-SIDE DECOMPRESSION IN LUSTRE



Anna Fuchs, Michael Kuhn

TODOs - I

Memory

- Client: correct release of pages
- Server: correct summary of used memory
- Compression pool
 - Atomic allocation of many blocks
 - Clean up forgotten blocks when freeing
 - Threshold for less memory than required
- Configuration
 - Compatibility version checks
 - Dynamic enabling/disabling of compression
 - Set correct page pool size
 - Get/set ZFS record size for chunk size
 - Compression threshold
 - Algorithms and policies

TODOs - II

- Algorithms
 - Add more; also page-based
 - Define policy of mixing algos per file/RPC/stripe
 - Decision step
- Style, refactoring, etc.
- Add read path
- Testing
 - Build tests
 - Correctness tests
 - Memory tests
- RMW fix offsets (do not assume page aligned data)
- Error handling fallbacks, recovery, crash

Store compressed data

- Requires ZFS changes
 - Test our changes
 - Submit to ZFS
- Consider ZFS' and Lustre's community reviews
- Client mechanisms remain the same for client-server or end-to-end compression
- Continue implementing support for OSS, MDS and ZFS