Enhanced Adaptive Compression in Lustre

Anna Fuchs, Michael Kuhn, Julian Kunkel, Thomas Ludwig

anna.fuchs@informatik.uni-hamburg.de

https://wr.informatik.uni-hamburg.de/research/projects/ipcc-l/

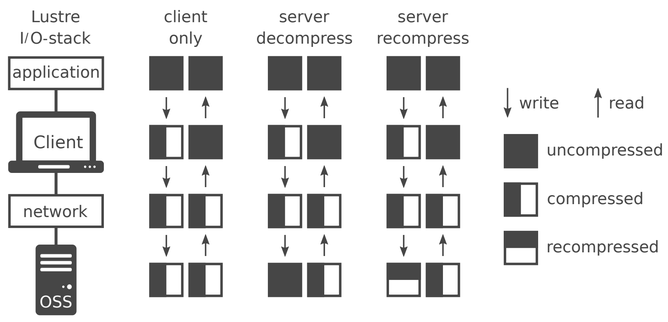
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Motivation

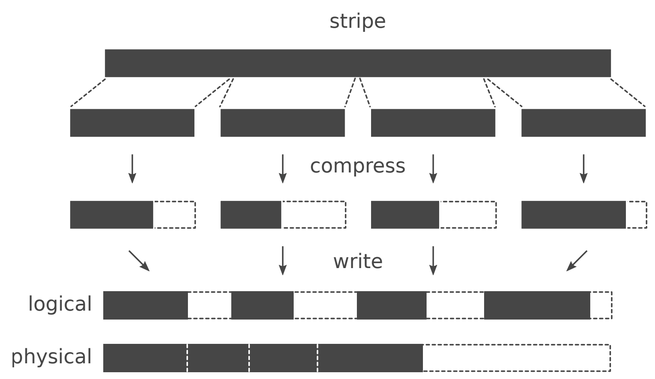
- Huge data amounts cause high storage costs
  - Can be reduced by compression
  - Up to 50% with no negative impact on capacity [1]
- CPU/Disk performance gap
  - Heavy I/O is a bottleneck
  - Use some computational resources to compensate
- High potential for Lustre file system
  - Client-side for higher throughput (less data to transfer)
  - Server-side for more capacity (less data to store)
  - Overall for higher efficiency at lower costs

Design

- Transparent compression based on stripes
- Possible on client and/or server, needs integration in backend



- Current focus on ZFS backend due to existing infrastructure
  - Pass through pre-compressed blocks with the logical and physical sizes
  - Use internal tree structure with additional metadata per record
  - Fits into compressed send/receive interface
  - Aiming for compatibility to access data directly from ZFS
- Stripes divided into chunks
  - Allows for parallel compression
  - Aligning to ZFS records for better read-ahead
  - Reduces read-modify-write overhead (ZFS-RMW for every record)



Challenges

- **Read-ahead**
  Problem
  - Naive compression causes gaps in continuous data stream
  - Read-ahead can not predict next chunk

  Solution
  - Gaps only logically; block structure continuous within ZFS-tree
  - Predicting data by physical size while showing logical size outwards
- **Read-modify-write**
  Problem
  - Can expand to read-decompress-modify-compress-write
  - Affects the complete compressed block (stripe)
  - Decompression, copy and moving of originally unaffected data

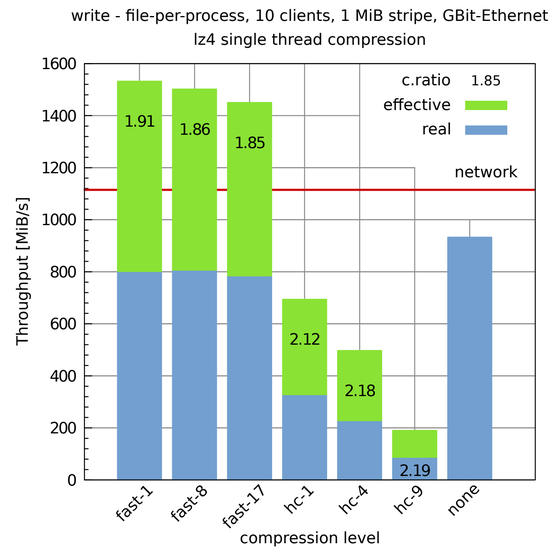
  Solution
  - Intelligent decisions when and where to compress (e.g. deactivate for random access or involve server for de-/compression)
  - Smaller independent chunks reduce negative impact on adjacent data within a stripe (decompression of a chunk, but not complete stripe)
  - Due to tree stuctured blocks no expensive moving of neighbored blocks

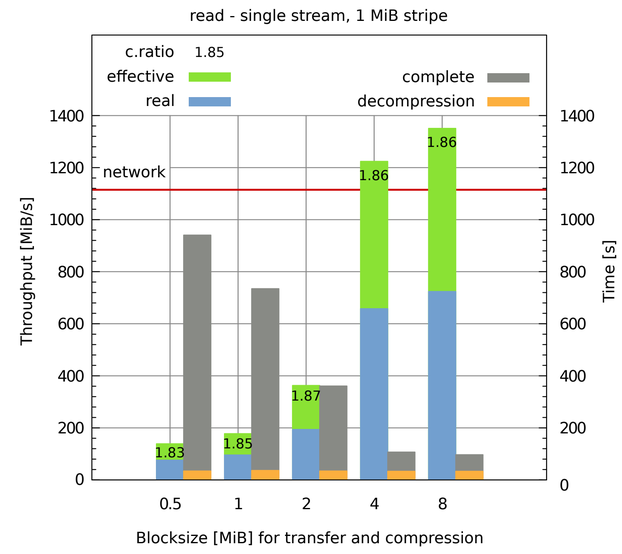
Evaluation

- Suitable compression algorithms (speed in MB/s) [1]
  - Compression (c.) speed faster than some networks
  - Very fast decompression (d.)

algo	c.speed	d.speed	ratio
lz4	1,796	5,178	1.923
lz4fast	2,945	6,460	1.825
lz4hc	258	4,333	2.0
zstd	658	2,019	2.326

- Userspace simulation





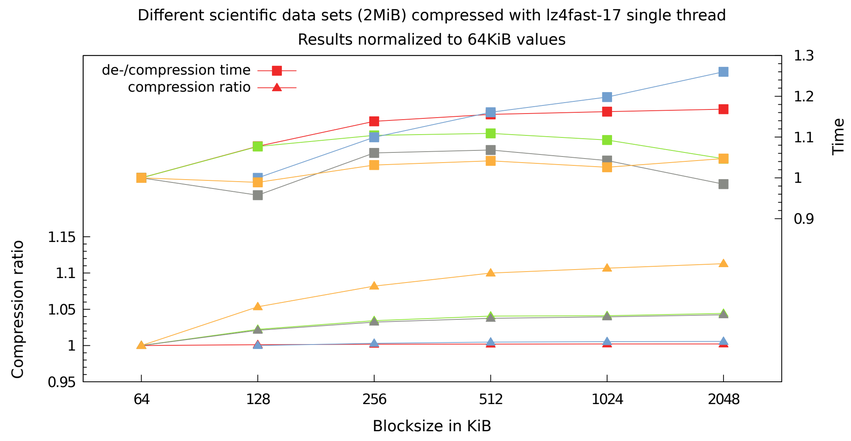
- Compression enables higher effective network throughput
- Read-ahead problems with naive compression
- Many small blocks with gaps drop performance
- Single thread compression insufficient for fast networks
  * Simplified calculations of profitability depending on network speed

$$c.speed - \frac{c.speed}{c.ratio} > network\ speed \Rightarrow c.speed > \frac{network\ speed}{1 - \frac{1}{c.ratio}}$$

  * Example: single thread lz4fast non-overlapping compression with network data rate of 1,500 MB/s

$$2,945\ MB/s > \frac{1,500\ MB/s}{1 - \frac{1}{1.825}} \Rightarrow 2,945\ MB/s \not> 3318\ MB/s$$

- Acceptable ratios and times with smaller blocks
  - ZFS records (min. 128KB) are still suitable for efficient compression



Future Work

- ZFS/Lustre integration (interface, chunk metadata, buffer usage)
- Benefits for other projects
  - Update lz4 and introduce new lz4fast in kernel (4.11) and ZFS
  - Extend ZFS compression features
- Adaptive compression, dynamic decisions
  - Consider static and dynamic system metrics (system config., load, etc.)
  - High-level user hints via ladvice (access patterns, data structure, etc.)
  - Internal decision possible for each chunk for highest efficiency

References

[1] Michael Kuhn, Julian Kunkel, and Thomas Ludwig. Data Compression for Climate Data. *Supercomputing Frontiers and Innovations*, pages 75–94, 06 2016.

Acknowledgements

We thank Intel Corporation for their support and funding our Intel Parallel Computing Center to integrate enhanced adaptive compression into Lustre.