Enhanced Adaptive Compression in Lustre

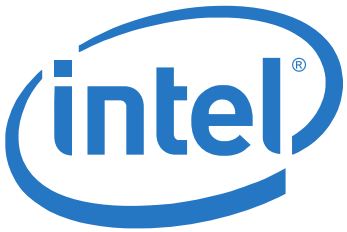
IPCC for Lustre

Anna Fuchs

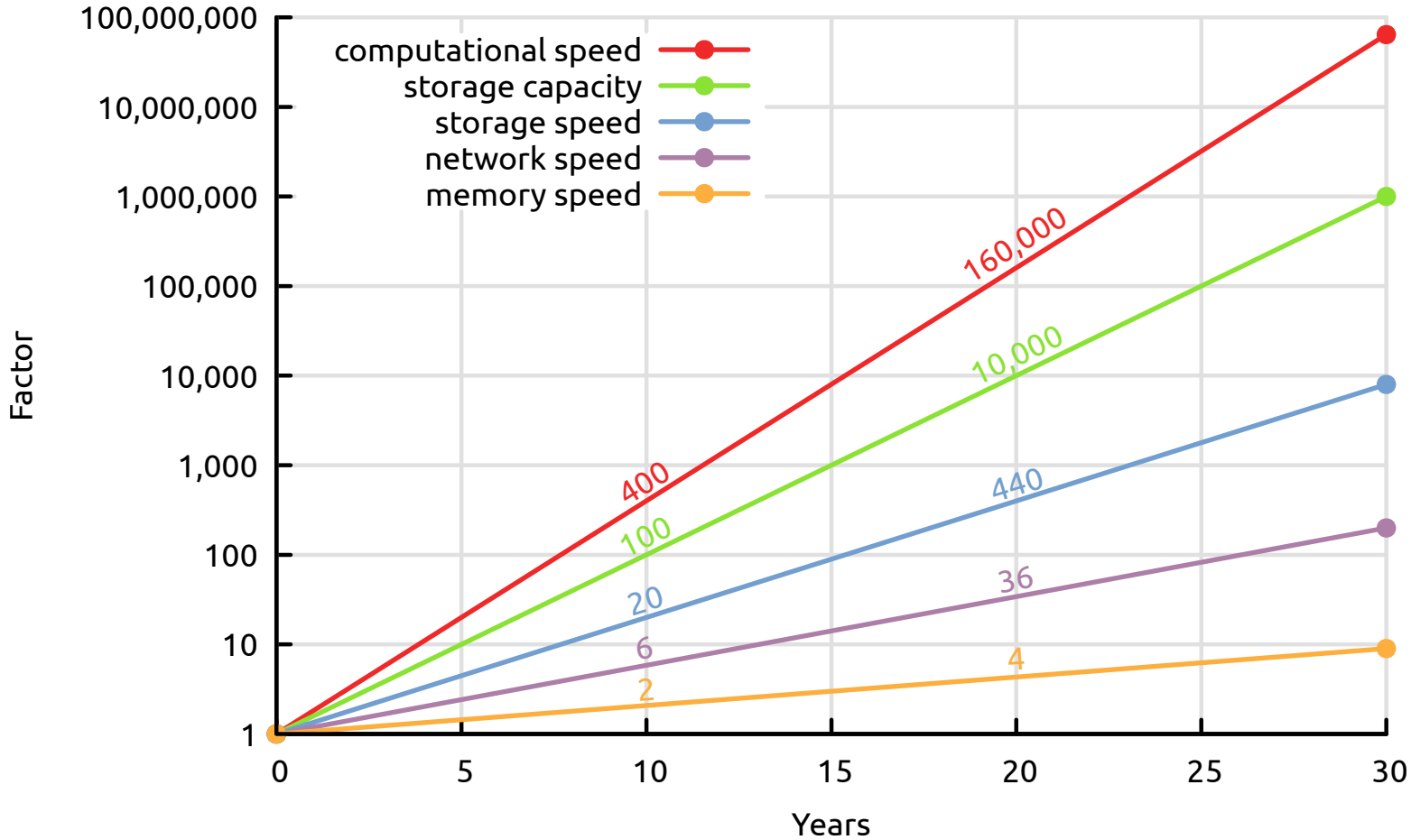
anna.fuchs@informatik.uni-hamburg.de

Research Group Scientific Computing
Department of Informatics, Universität Hamburg

2017-06-20

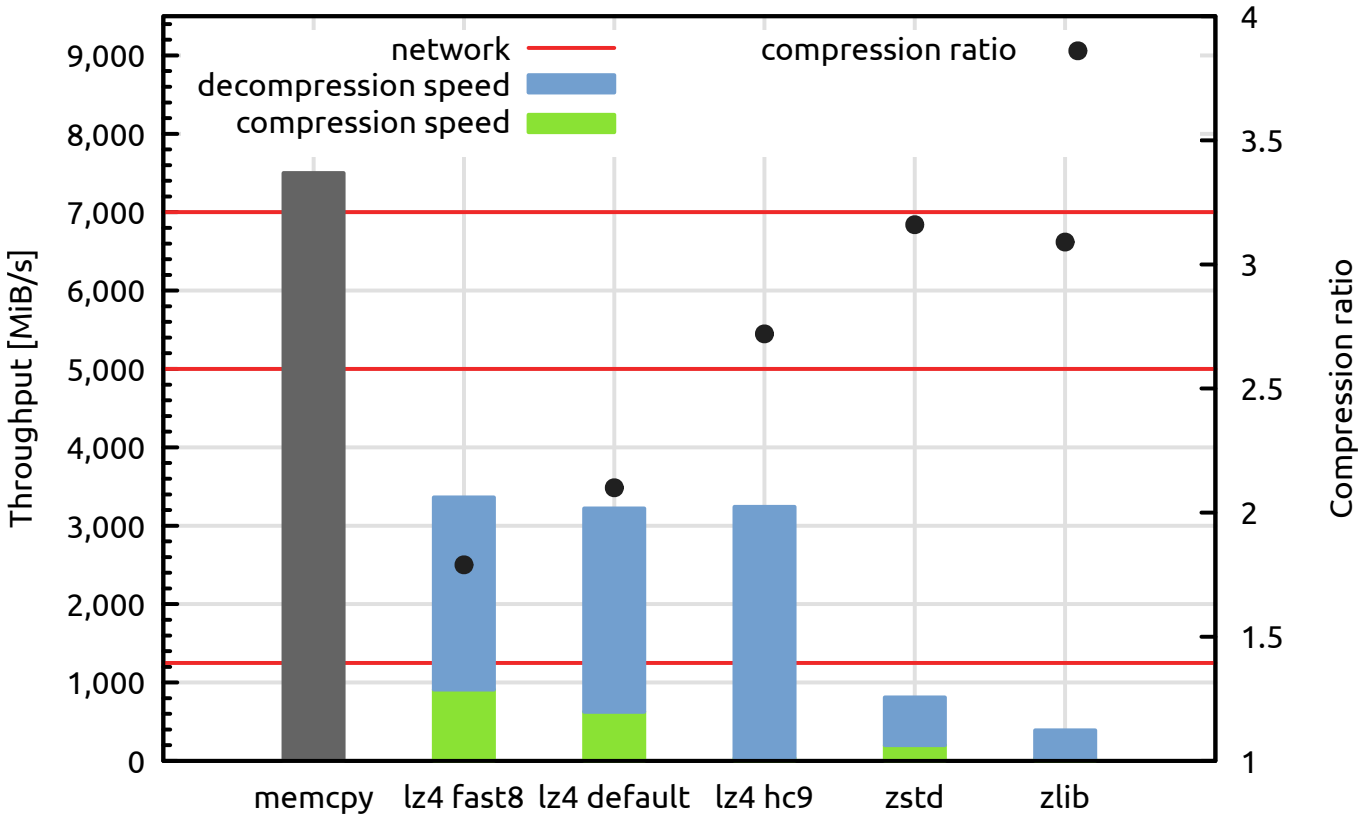


Problem description



- Growing gap between technologies; I/O is a bottleneck
- Use computational power to compensate other deficits
- Compression can reduce storage costs and improve throughput

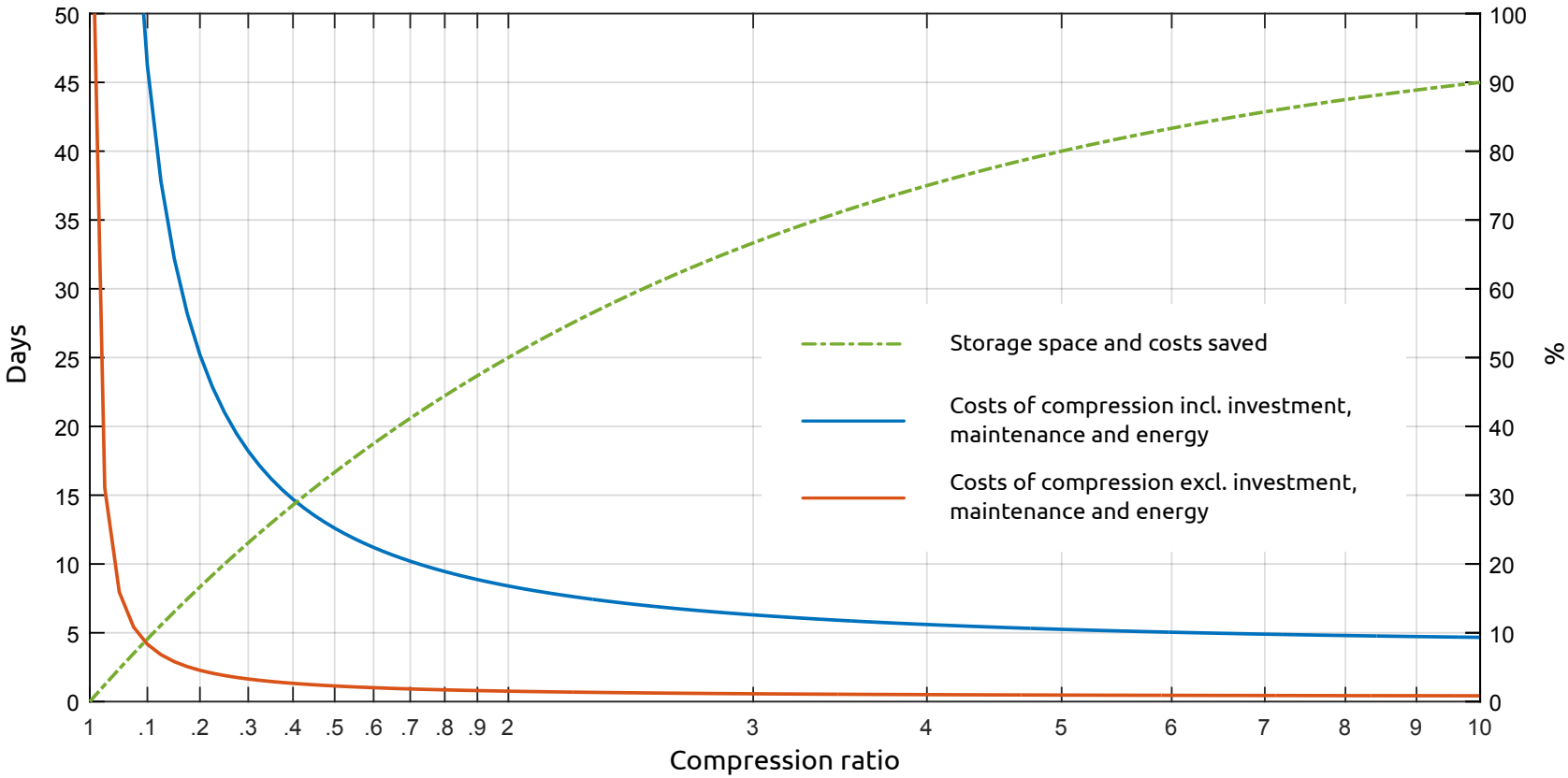
Compression algorithms



- Single stream still insufficient with faster networks
- Simplified: profitability depending on network speed

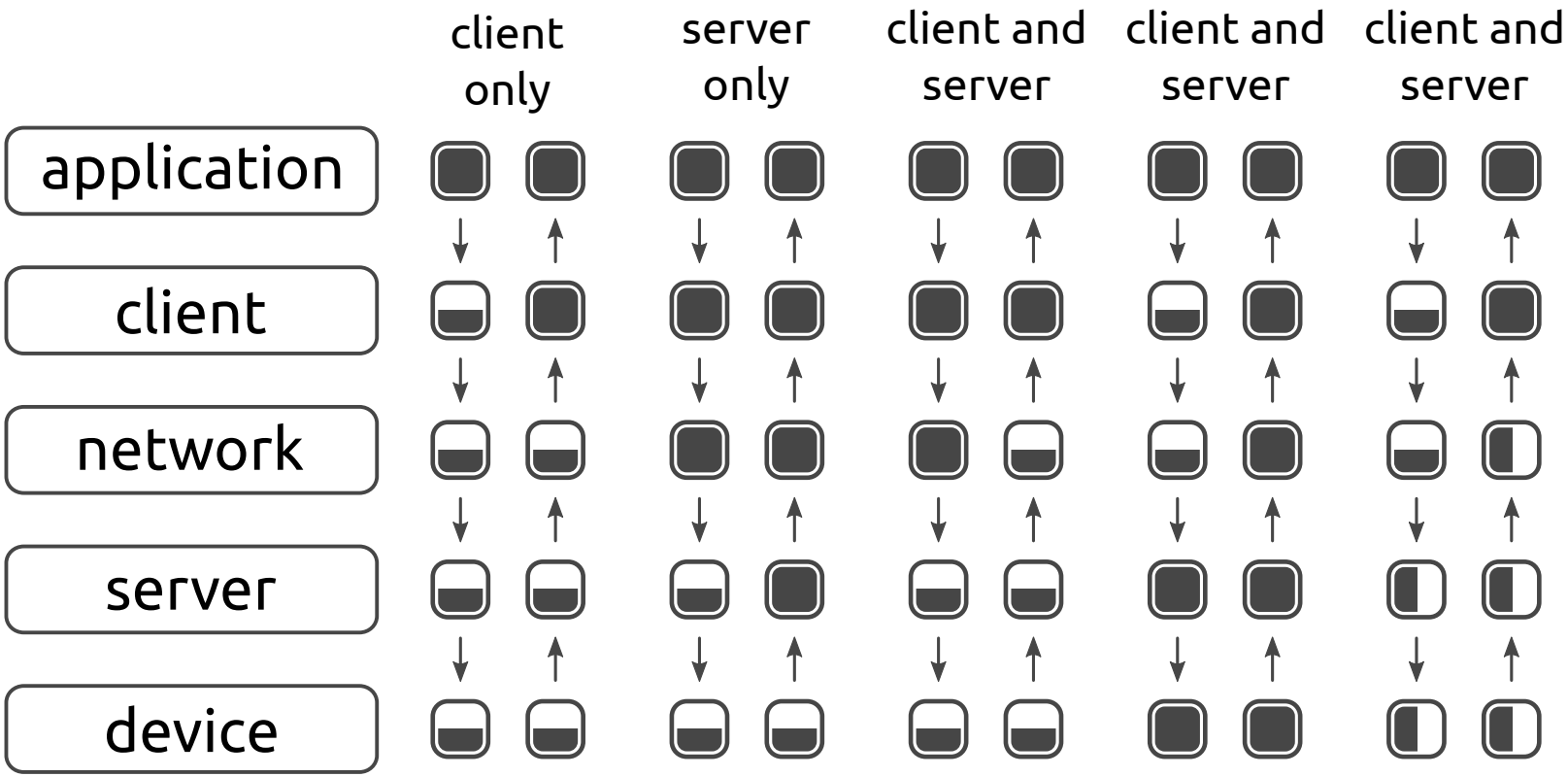
$$c.speed - \frac{c.speed}{c.ratio} > network\ speed \Rightarrow c.speed > \frac{network\ speed}{1 - \frac{1}{c.ratio}}$$

Compression benefits



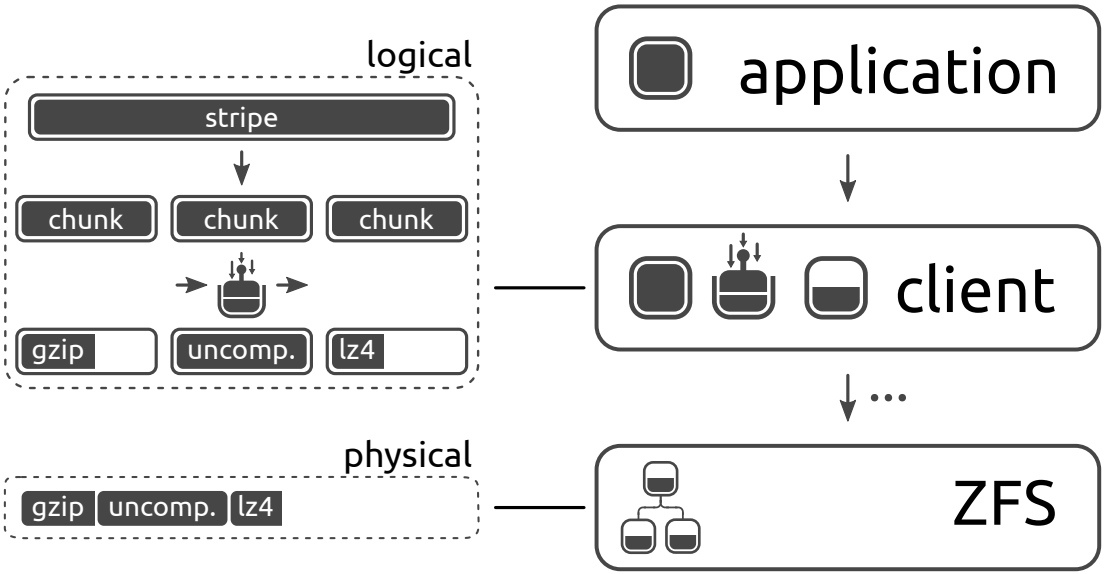
- Optimal use of compression pays off after couple of days
- Estimated mean residence time of data on servers > compression amortization time

Infrastructure



- Network benefits from client-side compression
- Focus on ZFS due to its built-in support for compression

Client



- Parallel compression of chunks on the client
- Mix compression per chunk is possible
- Logical (and uncompressed) chunks aligned to ZFS's records
- Still sufficient compression ratio with smaller chunks (lz fast)
  - 8 x 128 KiB chunks compared to 1 MiB stripe
  - ≈5% lower ratio and ≈5% faster compression

Challenges

- Readahead
  - Gaps caused by compression prevent readahead
  - Smart integration into block structure of ZFS to solve it
- Read-modify-write
  - Expands to read-decompress-modify-compress-write
  - Intensive case study to determine common access patterns
  - Optimizations and balancing of block sizes and client and/or server de-/compression
- Compatibility to ZFS
  - Reuse ZFS's infrastructure as much as possible
  - Allow ZFS decompress externally compressed data
  - Enable correct data handling without Lustre

Goals

- Make compression a universal feature of complete Lustre stack
- Enabling by default without performance degradation
- Transparent but controllable
- Adaptive decisions or explicit control
  - Whether: yes/no
  - Where: client/server
  - How: algorithm
  - When: threshold of data size
  - What: specific files or directories
  - ...
- Take system's state into account
  - Static configuration: network speed, number of clients/servers/CPUs, etc.
  - Dynamic: system load, input data, etc.

Project

- Research project – IPCC-L Hamburg
- Schedule June 2016 – January 2019
- Currently working on infrastructure
- Intel support, first prototype to be published soon
- Contributions and related work
  - lz4 in kernel
  - ZFS autocompression PR pending
  - ZFS infrastructure
  - Community spread in academia

- Welcome to meet and talk to us
- anna.fuchs@informatik.uni-hamburg.de
- Research poster at ISC'17 [Substanz 1+2, Forum]
- Student Cluster Competition [A-1452] Universität Hamburg