Enhanced Adaptive Compression in Lustre

Michael Kuhn

Research Group Scientific Computing
Department of Informatics
Universität Hamburg

2016-03-09
About Us: Scientific Computing

- Analysis of parallel I/O
- I/O & energy tracing tools
- Middleware optimization
- Alternative I/O interfaces
- Data reduction techniques
- Cost & energy efficiency
1 Motivation

2 Advanced compression

3 Conclusion
Gap between computation and storage

- Capacity and performance continue to increase exponentially
  - Different components improve at different speeds
- I/O is becoming an increasingly important problem
  - Data can be produced faster but it becomes harder to store it
- Consequence: Spend more money on storage
  - Results in less available money for computation
  - Or more expensive systems overall
- Storage becomes a considerable portion of the TCO
  - DKRZ: $8,500 \times 10 \text{ W} = 85 \text{ kW} \approx 110,000 \text{ €} \text{ per year}$
Gap between computation and storage...

Figure: Development of CPU speed, HDD capacity and HDD speed

- Processor speed: 400x every ten years (based on TOP500)
- Disk capacity: 100x every ten years
- Disk speed: 20x every ten years
Overview

- Compression in the file system can already be used today
  - Lustre supports ZFS backend
  - Turn on compression in ZFS
- Currently only static approaches for compression
  - One compression algorithm per file system
  - We would like to use a more dynamic approach
- Use semantical information to improve compression
  - Even adaptive compression needs to guess
  - More efficient application-specific compression
Overview...

**Figure:** Lustre architecture with ZFS compression
Feature Wishlist

- **Properly support compression in the file system**
  - Make it an actual feature
  - Interaction with application-specific compression

- **Allow developers to specify useful information**
  - Additional knowledge about data (variance, patterns etc.)
  - Leverage semantical information across the whole stack

- **Provide data reduction at a central layer**
  - Currently, all layers implement their own solutions
  - Redundant operations, wrong ordering etc.
File system support

- Support desirable at different levels
  - On servers, clients and within applications
- Each has advantages and disadvantages
  - Compression on the client influences computation but can save network bandwidth
File system support...

**Application**
- compressed
  - + No app. modifications
  - - CPU overhead (negligible)

**Network**
- compressed
  - + Network throughput inc.

**Storage**
- Lustre
  - compressed
    - + Proper interaction
    - - Lustre modifications
- ZFS
  - compressed
    - MDTs
    - OSSs
    - MDSs
    - OSTs
    - - CPU overhead (negligible)
    - - Energy overhead (~1%)

**Figure:** Lustre architecture with advanced compression support
File system support...

- Compression is not supported on the clients
  - Add support to Lustre’s client
  - Completely transparent to applications
  - Configurable via `ladvise`

- Compression is static
  - Add support for adaptive compression
  - Can use information about the data, the current load etc.
  - Useful on both the clients and servers
Adaptive compression

- Added support for adaptive compression to ZFS
  - Directly usable by Lustre
- Support for different modes
  - Such as performance, archival and energy
- Different heuristics to determine compression algorithm
  - Based on the file type or cost function
- All algorithms are tried for cost function
  - Best one is chosen for the next batch of operations
Adaptive compression...

**Figure:** System utilization compressing mixed file using gzip-1
Adaptive compression...

Figure: System utilization compressing mixed file using archive mode
Application Interaction

- ADIOS provides an expressive I/O interface
  - Abstract description of applications’ I/O using XML
- Extend to support advanced data reduction
- Already offers some helpful functionality
  - Data transformations
  - `adios_{start,stop}_calculation`
  - `adios_end_iteration`
Application Interaction...

- Extend with further semantical information
  - Compressibility etc.

```xml
<adios-config host-language="C">
  <adios-group name="checkpoint">
    <var name="matrix" type="double" dimensions="..." variance="low" transform="compression:performance"/>
  </adios-group>
</adios-config>
```

Listing 1: ADIOS extensions
Conclusion & Future Work

- Compression bears the potential to reduce the TCO significantly
  - Client memory and network utilization can also be reduced
  - Useful for data not compressed by the scientists explicitly
- Explore the benefits of adaptive compression
- Interfaces that enable more intelligent compression using semantical information