

Kompression in Lustre

Anna Fuchs

Wissenschaftliches Rechnen, Fachbereich Informatik
Universität Hamburg

08.07.2016



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG



Outline

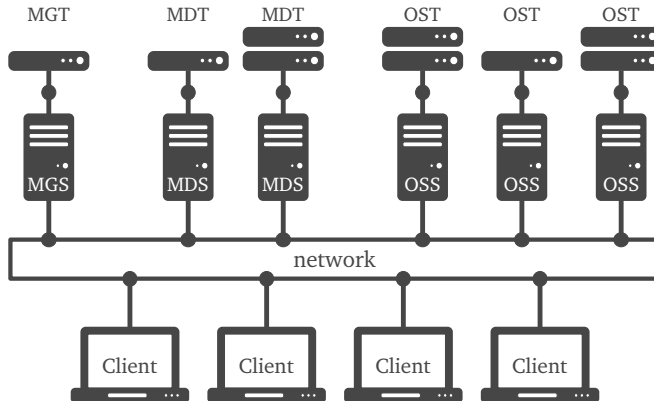
- 1 Einführung
- 2 Lustre
- 3 Kompression
- 4 Implementierung
- 5 Evaluation
- 6 Zusammenfassung

IPCC-L

- Intel Parallel Computing Center for Lustre "Enhanced Adaptive Compression in Lustre"
- E/A-Engpass mit Rechenleistung kompensieren
- Kompression kann Kosten sparen
 - Mehr Kapazität bzw. bessere Ausnutzung
 - Geringerer Stromverbrauch
 - Höherer Netzwerkdurchsatz

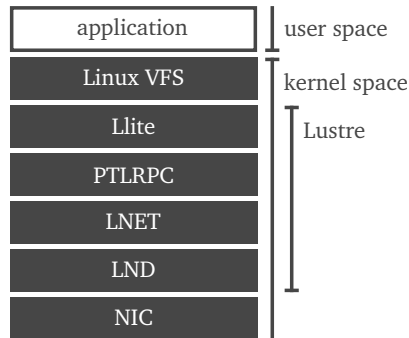
Architektur

- Lustre – paralleles verteiltes Dateisystem
- ZFS oder Idiskfs als Backend



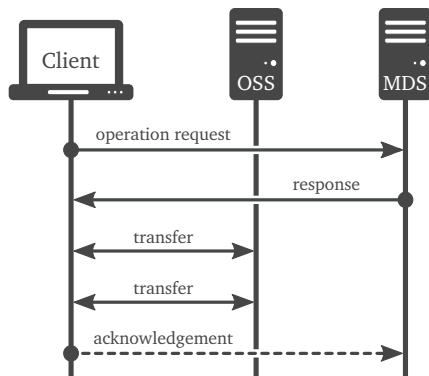
Softwarestack

- Llite
 - VFS-Kopplung mit POSIX-Interface
 - Eigentliche Lustre-Logik
- PTLRPC
 - Portal – Port-Identifikator
 - Remote Procedural Call – Interaktion zwischen Komponenten
- LNet
 - Lustre-Netzwerk



Datentransfer

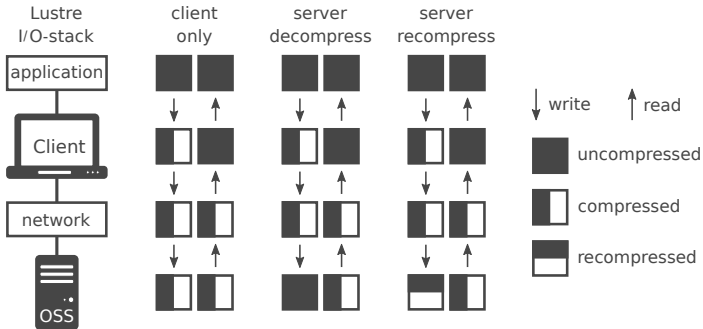
- RPC –
Kommunikationsgrundlage
zwischen allen Komponenten
 - Anfragen, Antworten,
Datentransfer
- Bulk – Datentransfer für größere
Datenmengen
 - $n \times \text{PAGE_SIZE}$
 - write/read
 - Alle Infos in einem Descriptor
packen



GSS

- Generic Security Service – Interface für Sicherheit und Verschlüsselung
- Schicht für Datenverschlüsselung innerhalb von PTLRPC
- Letzte Chance für Datentransformation vor dem Netzwerk
- Kein offiziell unterstütztes Feature; nicht stabil
- Keine persistente Verschlüsselung, nur Netzwerk
 - Client verschlüsselt, Server entschlüsselt
- Mechanismus – Implementierung der Algorithmen
- Angewendet auf RPC und Bulk-Daten

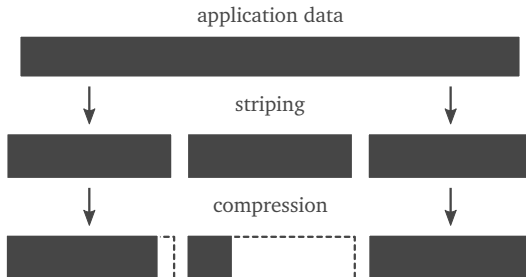
Lustre stack



- Unterschiedliche Kombinationen je nach Bedarf

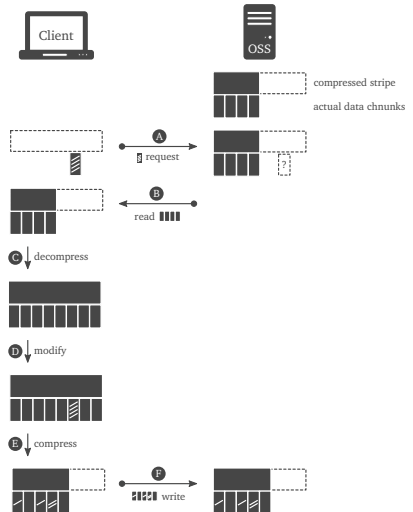
Grundfunktionalität

- Basierend auf Streifen
- Abbruch bei nicht komprimierbaren Daten
- Metadaten auf Basis von Streifen (oder Records)
- Innerhalb des PTLRPC-layer
 - GSS-layer Funktionalität wiederverwenden

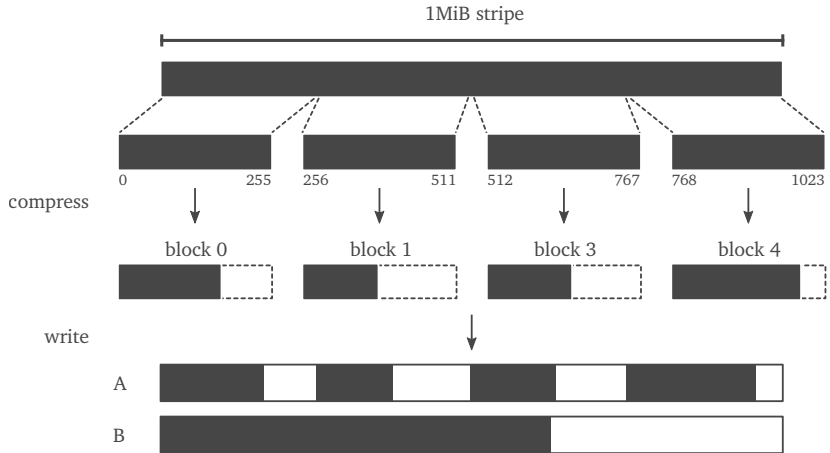


Leistungskritische Fälle

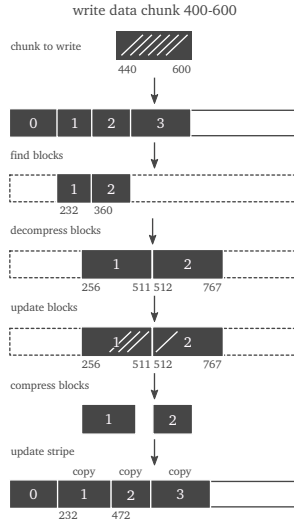
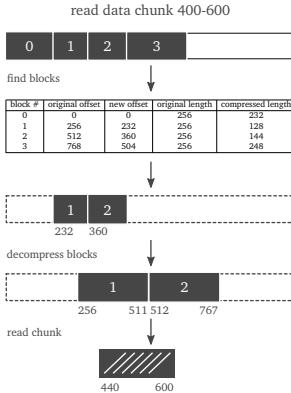
- Ausrichtung der komprimierten Streifen
- Lesen/Schreiben von Teilen der komprimierten Daten
- Mehrere Clients schreiben einen Streifen
- Read-modify-write → Read-decompress-modify-compress-write



Streifenteilung



Problematik Fall B



Features

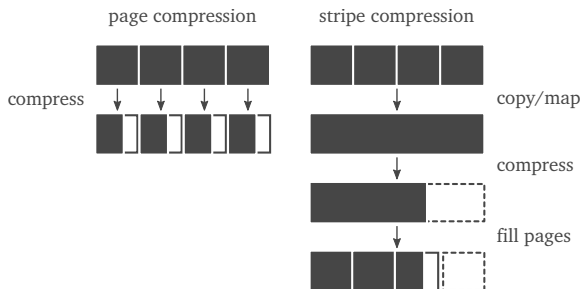
- Leistung
 - Verschiedene Algorithmen
 - Parallele Blockkompression (Threads)
 - Hardware-support für Kompression (Intel QuickAssist (QAT))
- ZFS
 - Kann Kompression
 - Infrastruktur wiederverwenden
 - Ausrichtungsprobleme vermeiden
 - Pointerbäume, um Lücken zu vermeiden
 - Zugriff per-record

Prototyp

- Innerhalb von PTLRPC, ein Schritt vor GSS
- Extra Service/Modul einführen
- Nur Bulk-Daten betroffen
- Verschlüsselung nachträglich möglich
- RPC-Anfrage muss aktualisiert werden (Seitenzahl, Bytes, Fragmente etc.)

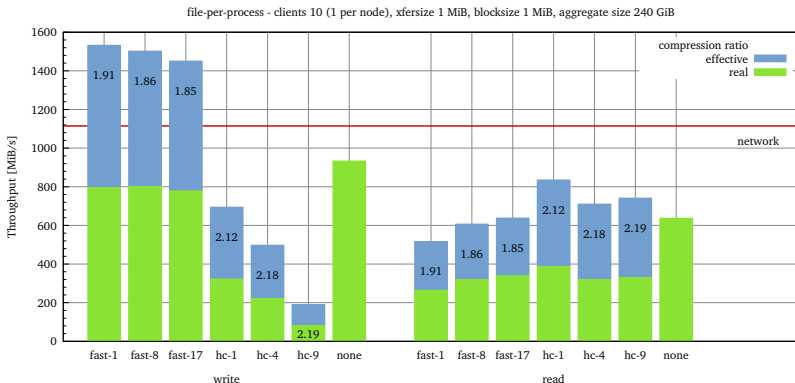
LZ4

- Nur Block-Format im Kernel implementiert (keine Frames)
- Originale Daten nur auf Seiten-Basis zugreifbar
- Aktuell sehr speicherintensive Kompression

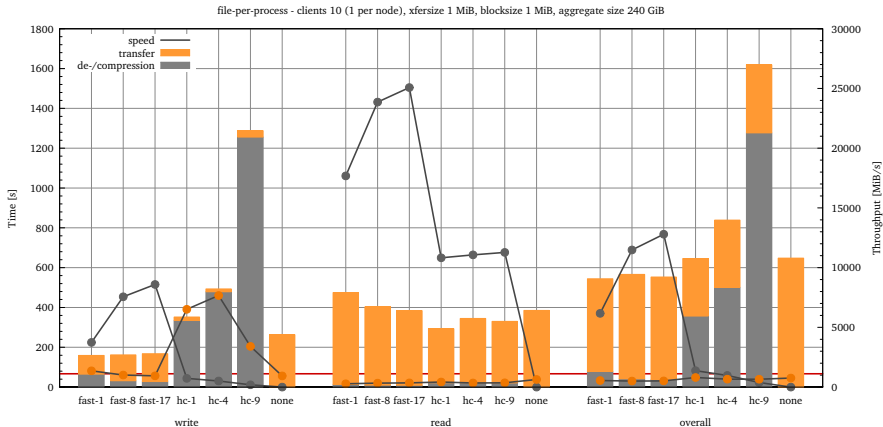


Userspace LZ4 mit IOR

- LZ4 – 3 GiB/s Kompression, 6,5 GiB/s Dekompression
- Modifiziertes IOR – LZ4 fast/HC mit 1-MiB Blöcken



Userspace LZ4 mit IOR ..



- Kompression vielversprechend
 - Durchsatzsteigerung
 - Kapazitätsgewinne
- Geeignete Algorithmen (schlecht geeignete Implementierung)
- Viele Features denkbar
- Leistung im Allgemeinen sehr abhängig vom System
- Noch viel zu tun