Report: ZFS on Linux Performance Evaluation

Norbert Schramm

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

2016-03-30

Agenda

Goal of the Project

- ZFS on Linux: Linux-Version of original ZFS by Sun
- stable-released in April 2013
- benefits against traditional file systems
- supported by Lustre since May 2013
- How good is ZFS on Linux Compared to other ZFS-Versions?

Comparing ZFS

- Tested ZFS on 3 Operating Systems
  - OpenIndiana (based on Illumos, Former Solaris)
  - FreeBSD
  - Linux (Ubuntu)
- Hardware
  - Core-i5 2500k (4x3.3 GHz), 8 GB RAM, 1 TB WD Black
  - Xeon E3110 (2x3.0 GHz), 8 GB RAM, 1 TB WD Black
  - Xeon X5677 (4x3.46 GHz), 32 GB RAM, 136 GB 15K SAS-Drive
- Benchmark
  - bonnie++

Comparing ZFS

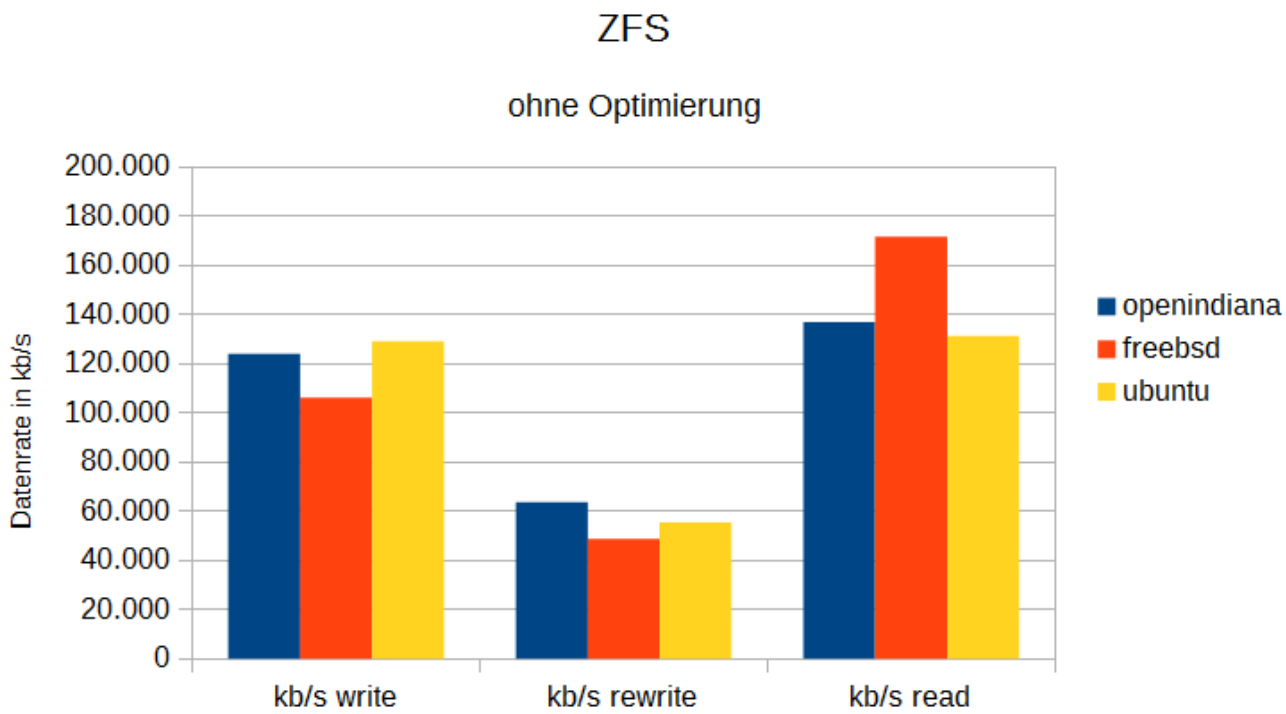


Figure: Core i5, bonnie++, ZFS initial

Comparing ZFS

- Ubuntu uses *relatime*
- also available on ZFS (initial: disabled)

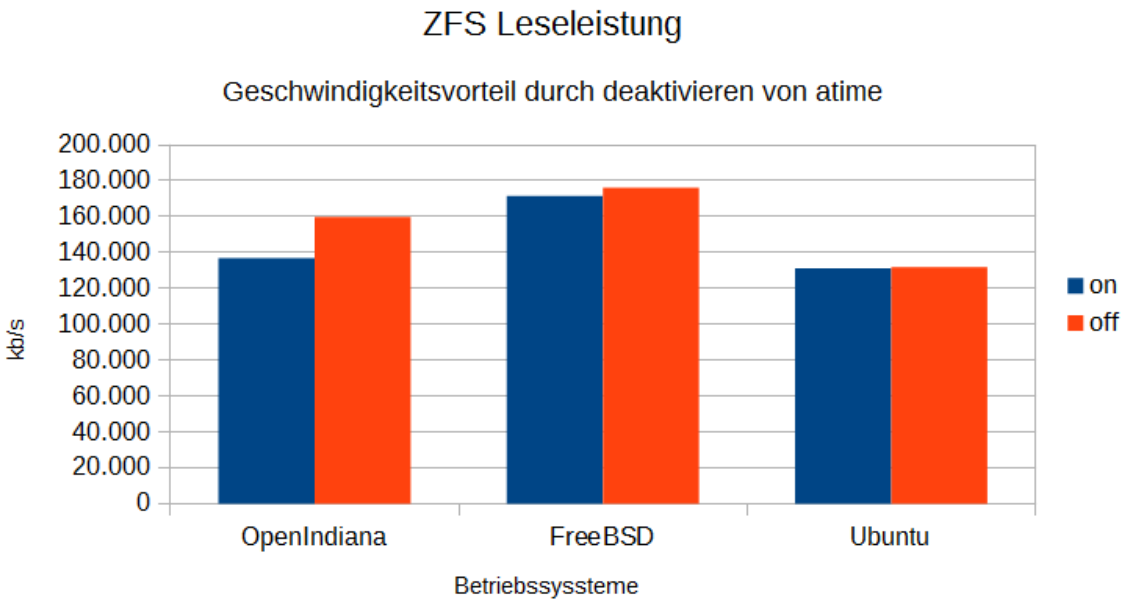


Figure: Core i5, atime optimisation

Comparing ZFS

- bonnie++ Data Compression Rate: > 130x

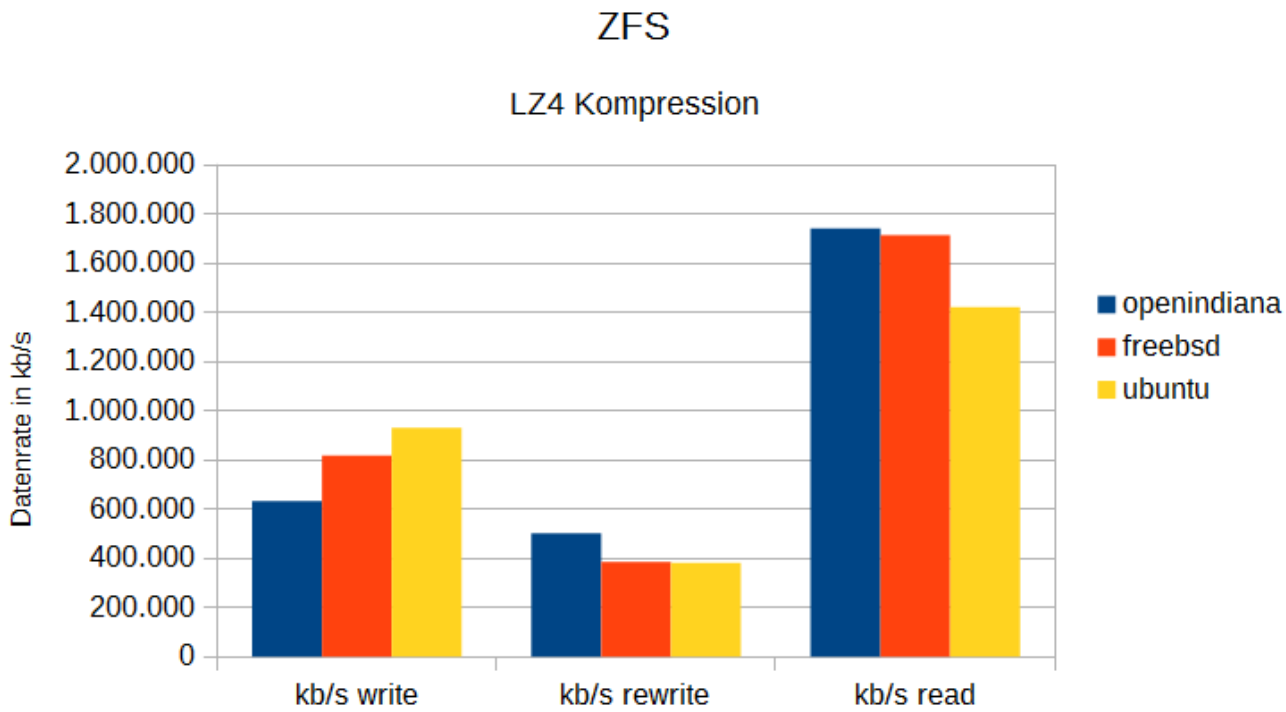


Figure: Core i5, LZ4 Compression

Comparing ZFS

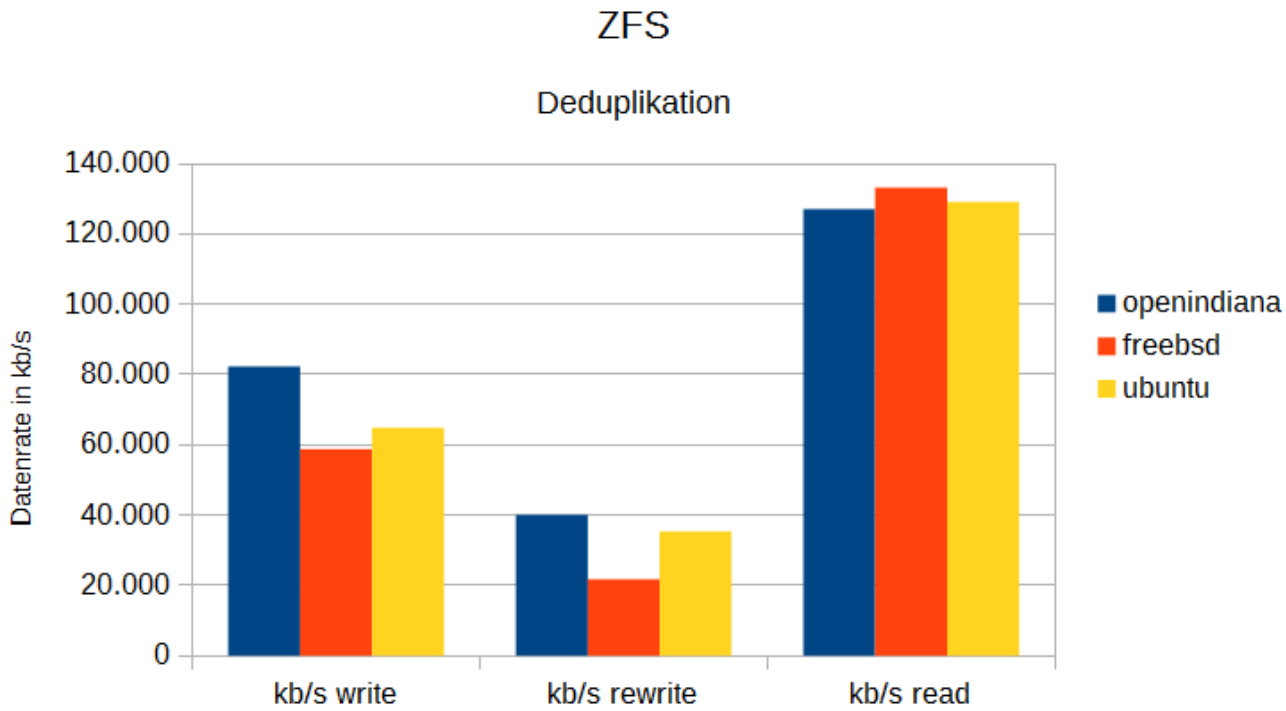


Figure: Core i5, Deduplication

Comparing ZFS

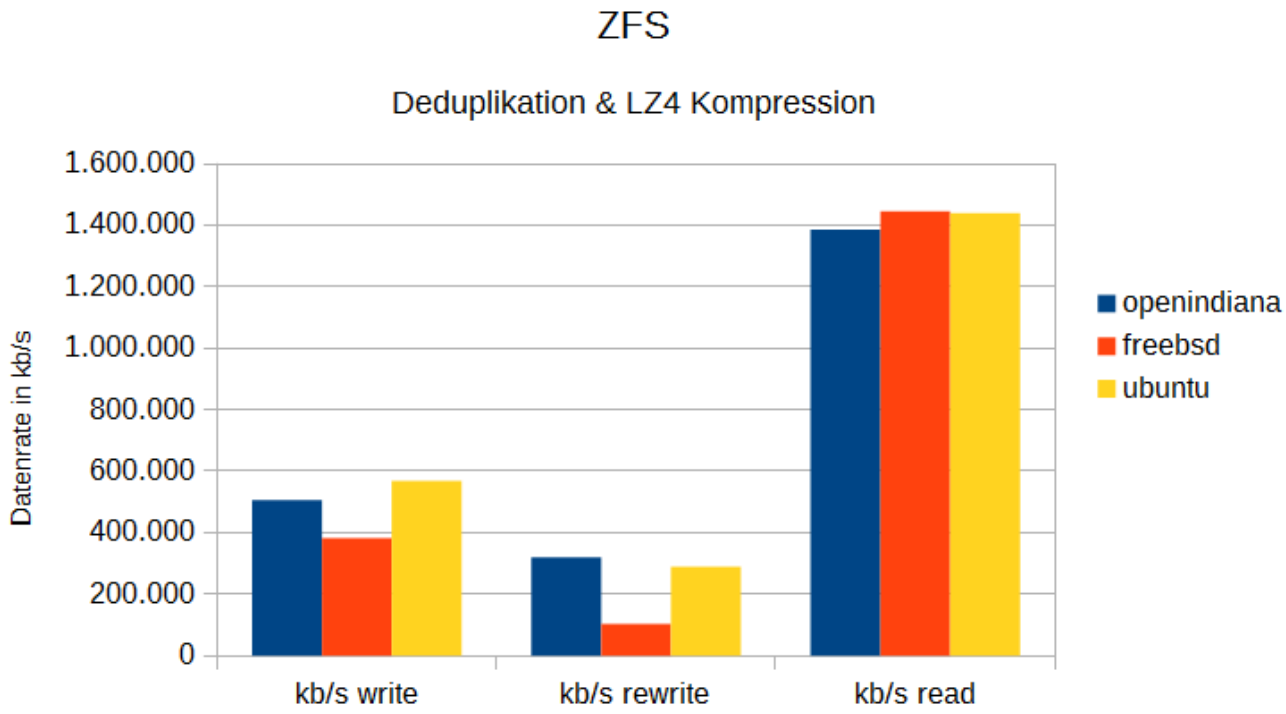


Figure: Core-i5, Compression % Deduplication

Comparing ZFS

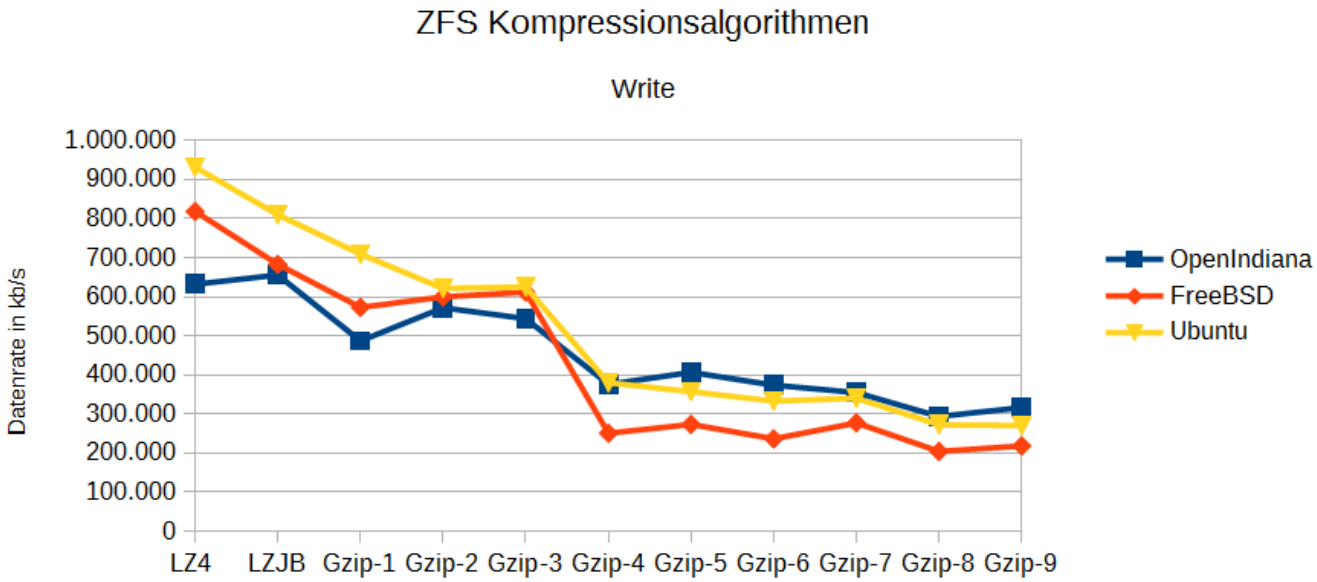


Figure: Core i5, Comparison Write

Comparing ZFS

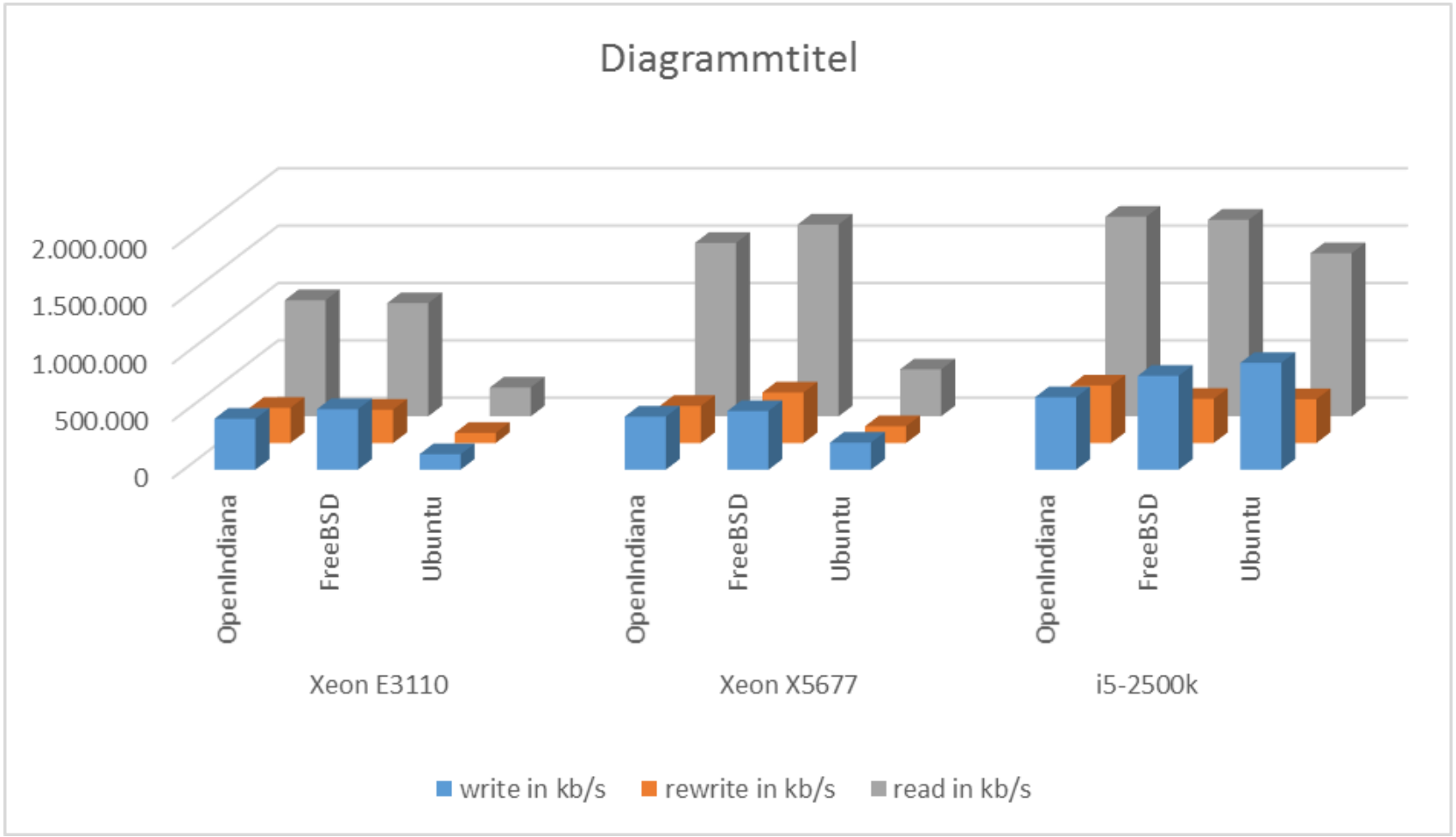


Figure: Comparison 3 Processor Generations with LZ4 Compression

ZoL Analysis

- recompiled Linux-Kernel with LOCK_STAT
- Using wrstat for detailed analysis
  - lock_stat
  - oprofile (debug kernel needed)
  - /proc file system
- analyzed run of bonnie++

Example: ZFS no Compression/Dedup

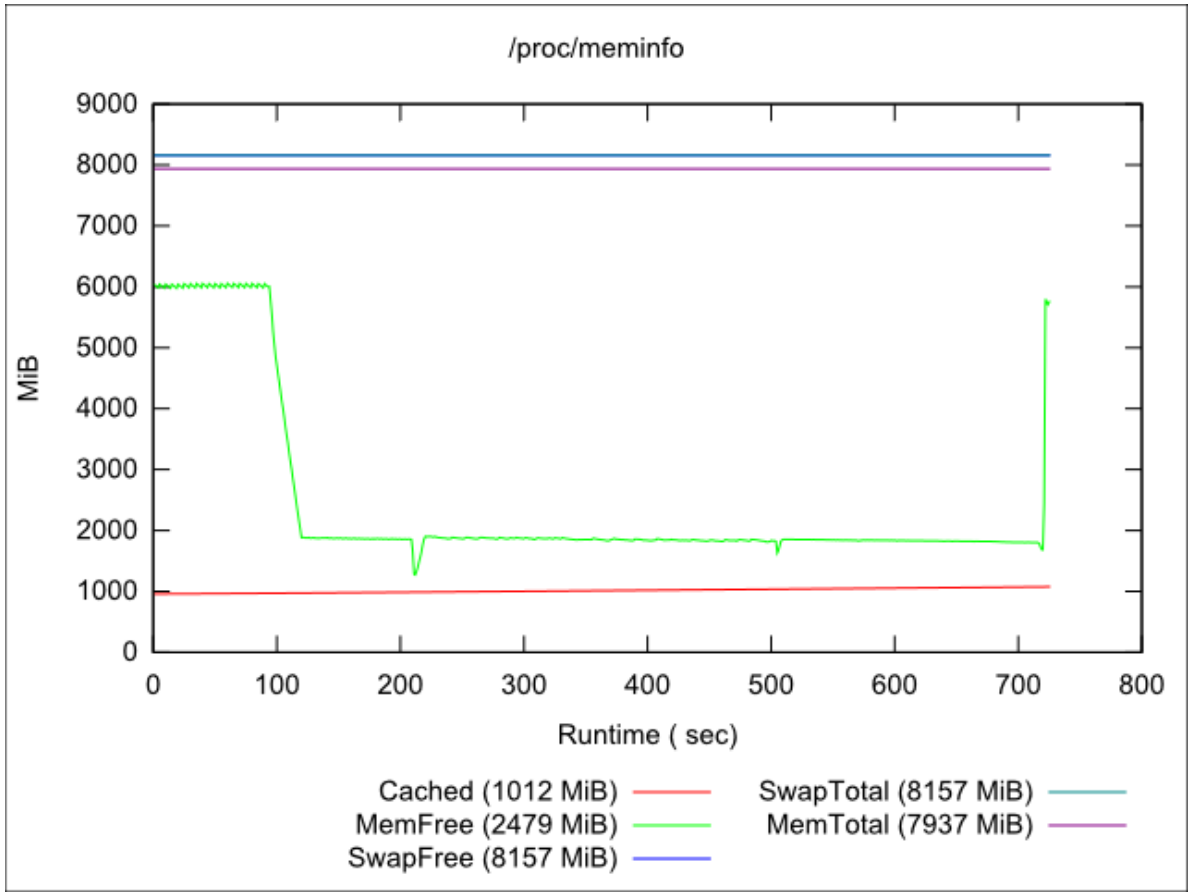


Figure: ZFS initial: RAM Usage

Example: ZFS no Compression/Dedup

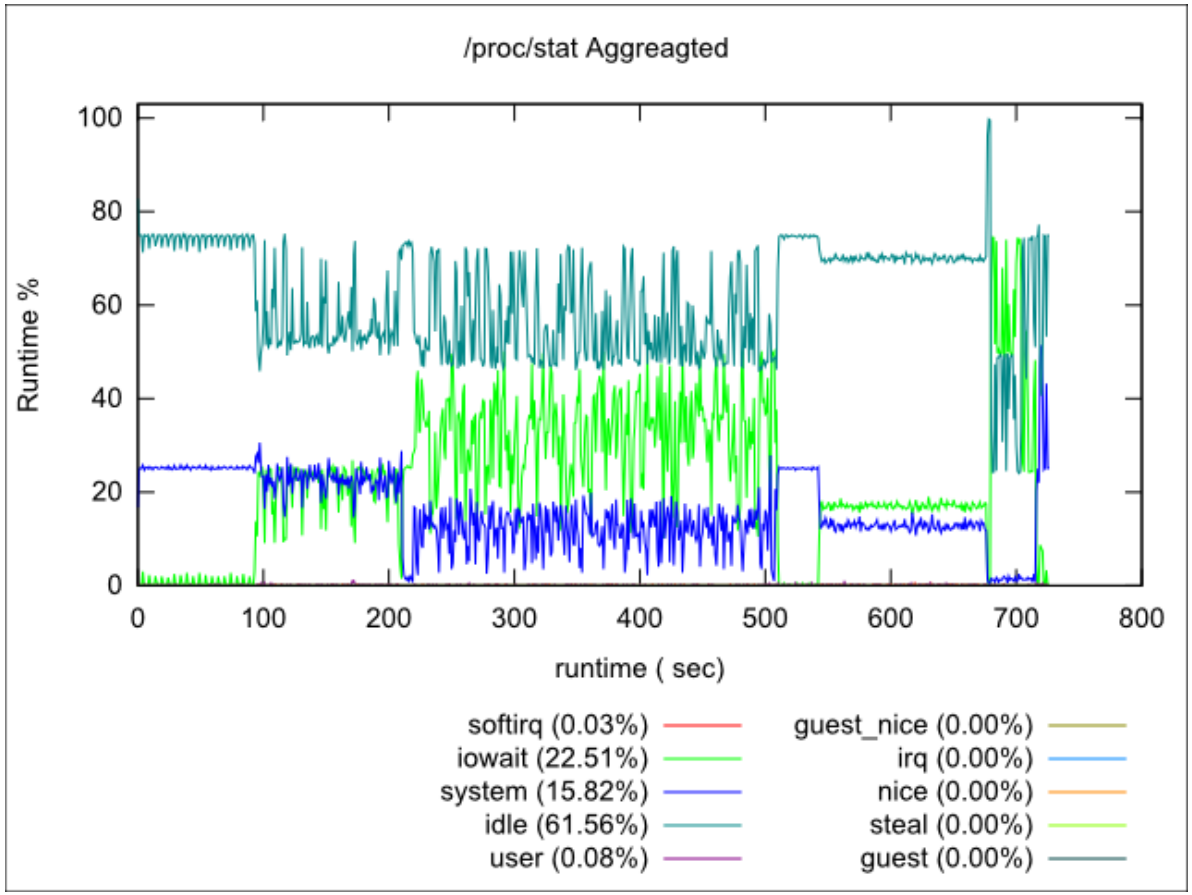


Figure: ZFS initial: CPU States

Example: ZFS no Compression/Dedup

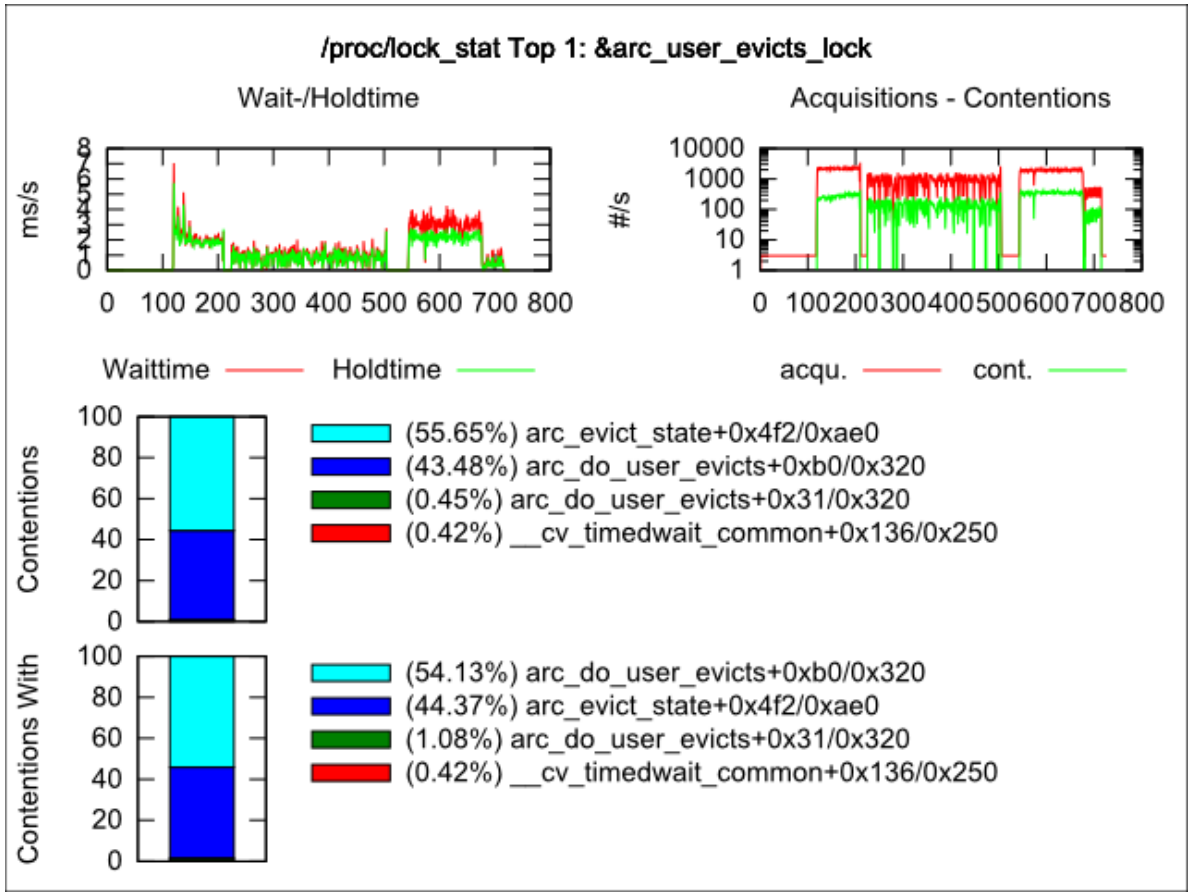


Figure: ZFS initial: Locks, when ARC is saturated

Lustre on ZFS: Preparation

- Test Setup
  - 3 Nodes
  - GBit Uplink to central Switch
  - Xeon X5560 4(8)x2.8 GHz, 12 GB RAM, 2TB HDD for ZFS
  - nehalem1: MGS, MDS, OSS
  - nehalem2, nehalem3: OSS
- Benchmark
  - serial: bonnie++
  - parallel: ior (with mpi)
- Operating System
  - CentOS 7

Lustre on ZFS: bonnie++

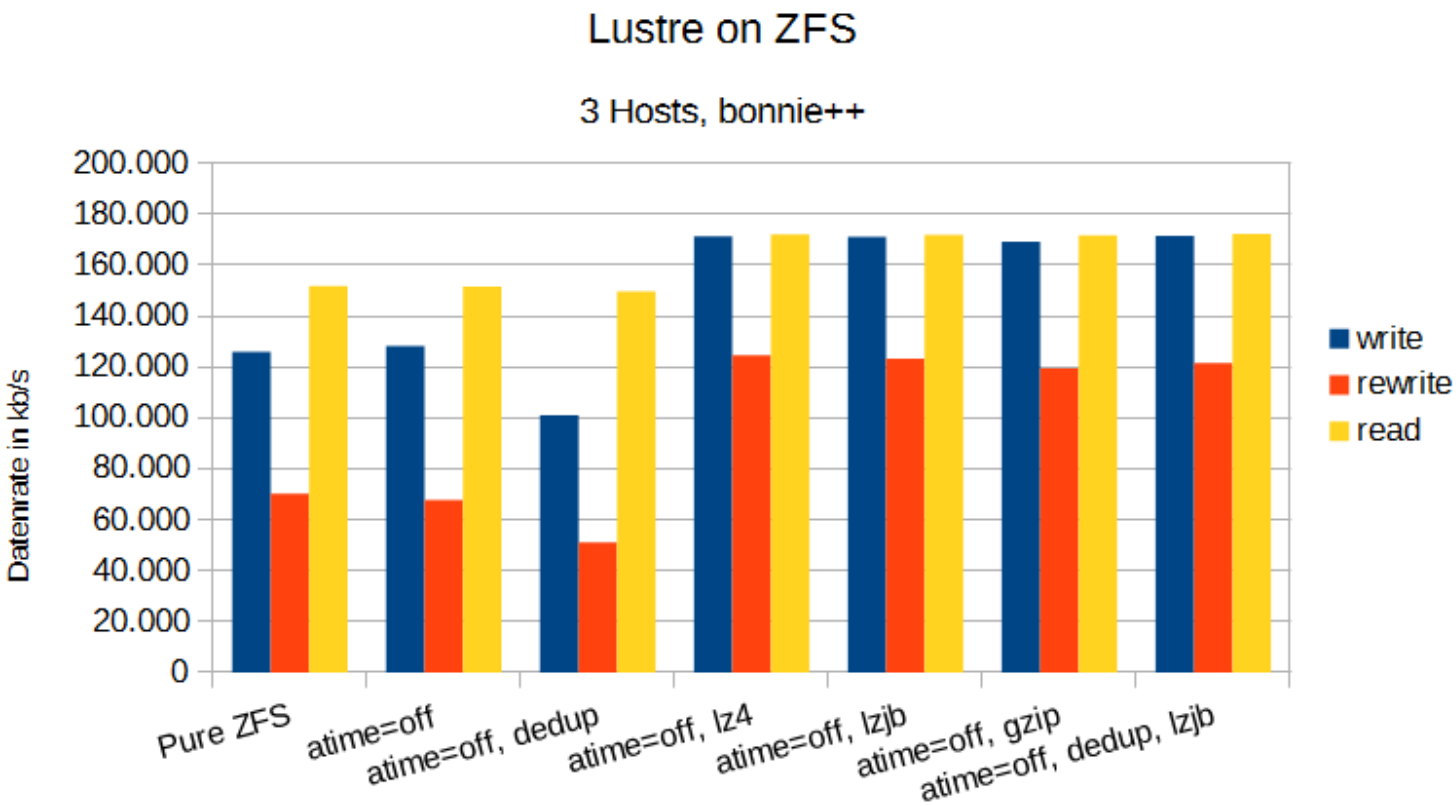


Figure: Bonnie++ Benchmark on Lustre on ZFS-Basis

Lustre on ZFS: bonnie++

- Problem: Data transfered uncompressed over the Network

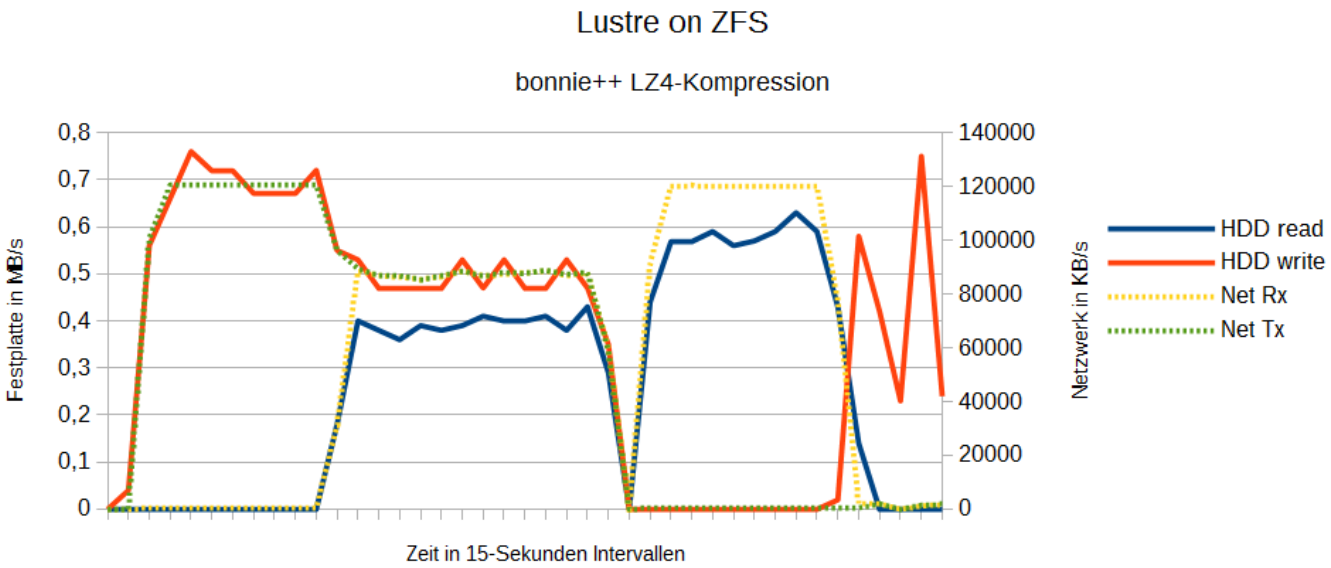
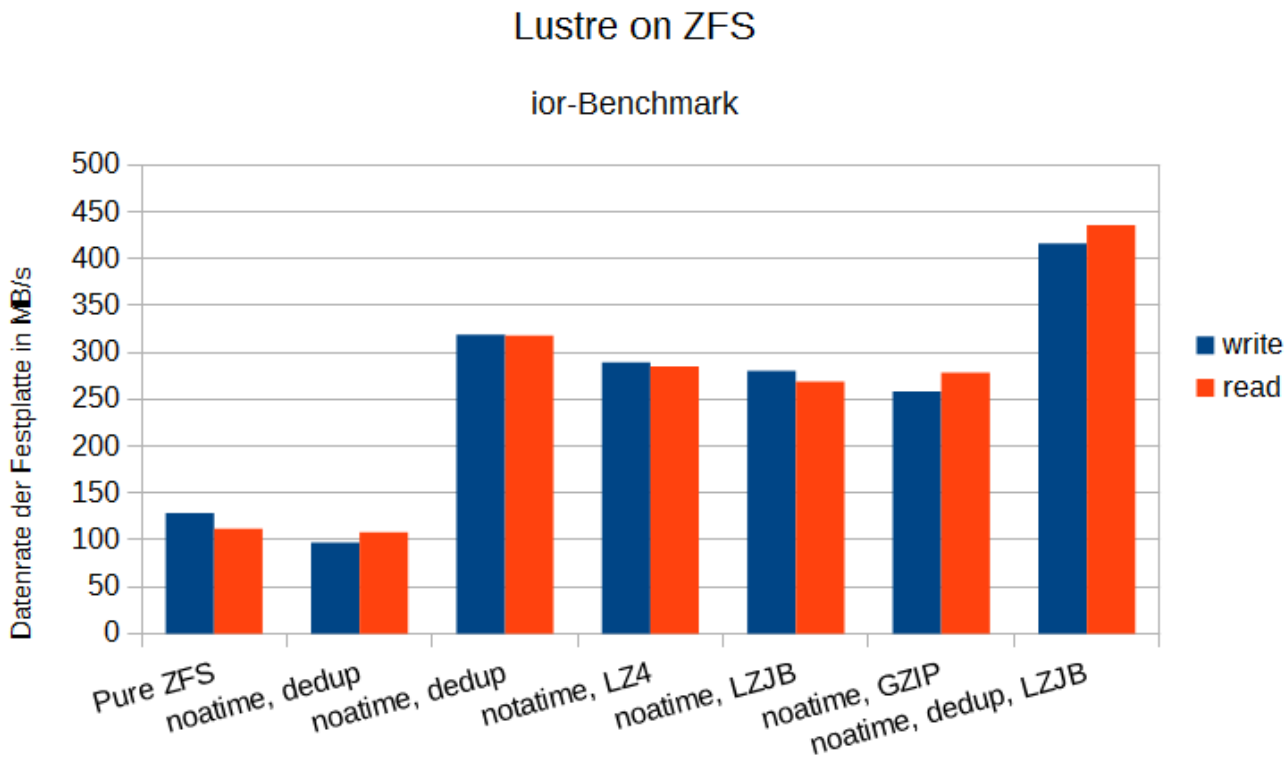


Figure: Network Limiting Problem

Lustre on ZFS: ior

- 6x ior (2 of each node)
- Dedup very good
- LZ4 fastests, Gzip slowest



Lustre on ZFS: ior

- Bursty Traffic when ARC is full
- Compressratio of ior-data: ca. 3.9x
  - more work on HDD

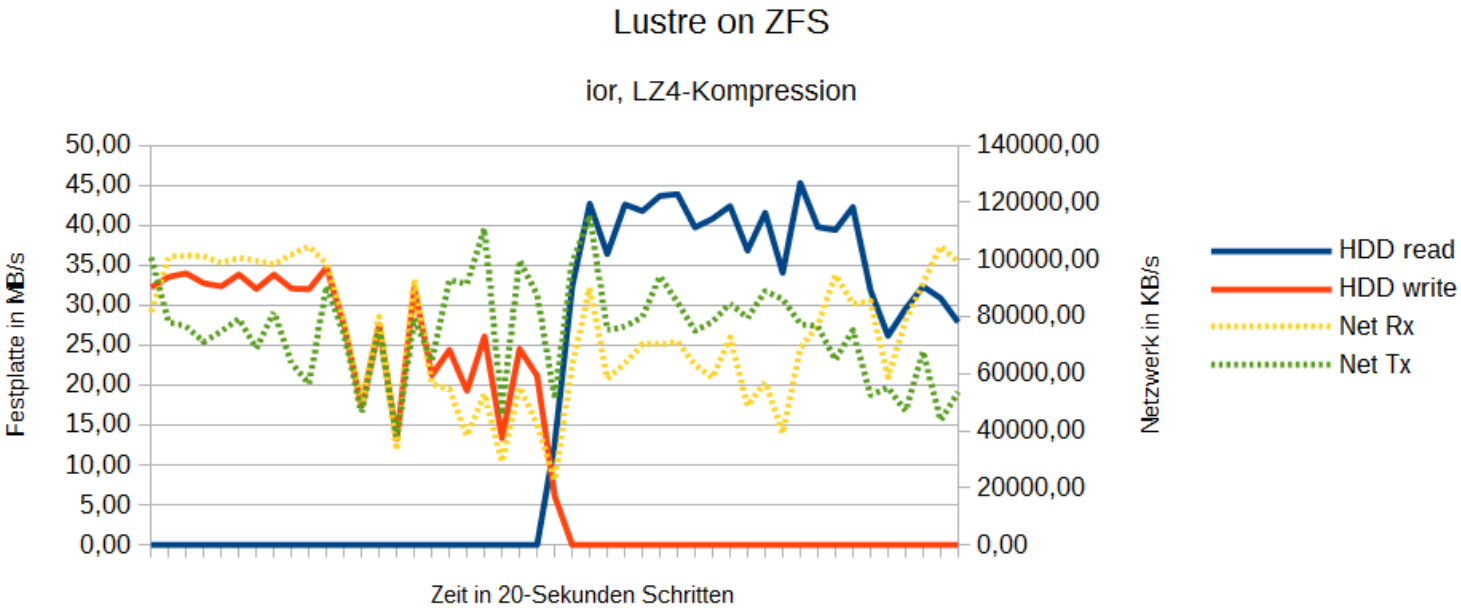


Figure: Network and HDD on nehalem1 while ior-run with ZFS-Compression

Lustre on ZFS: ior

- slow write performance thus parallel access
- txg_sync while no data is transfered
  - ZIL is located on disk, slows disk down

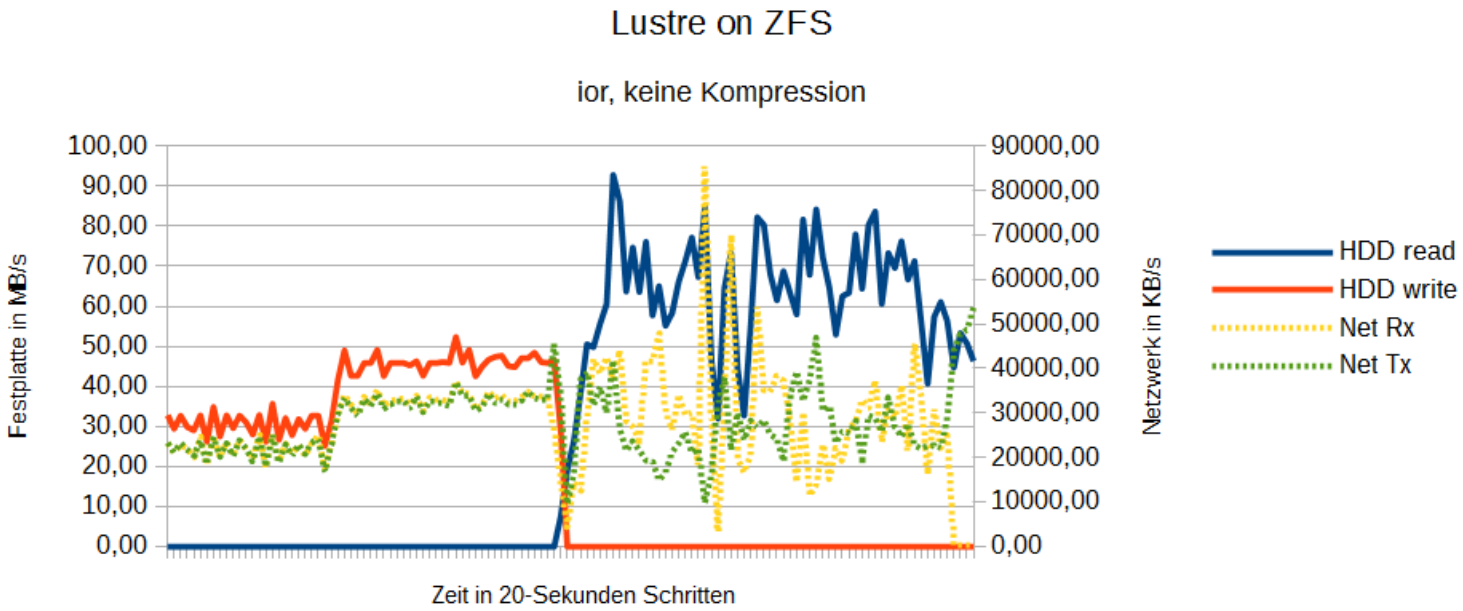


Figure: Network and HDD on nehalem1 while ior-run without ZFS-Compression

- ZoL is comparable to other versions (functionality, performance)
- actual Version: 0.6.5.6
- Version 1.0 when zvols are implementet
- Uses a lot of Locks while caching, can possibly slow down
- separate Device for L2ARC and SLOG can improve performance
- Lustre can get benefits from ZFS (e.g. Compression)

Thank you for your attention.
Questions?