Lustre at DKRZ

Julian M. Kunkel, Carsten Beyer, Olaf Gellert,
Hendryk Bockelmann, Eugen Betke

kunkel@dkrz.de

German Climate Computing Center (DKRZ)

2017-06-21

Lustre at DKRZ

- The Mistral supercomputer was shipped with Lustre
  - 4 PFLOP/s peak system, 3361 nodes, 102k cores
  - 52 PiB Lustre storage
  - Roughly 6 M EURO
  - See: `https://www.vi4io.org/hpsl/2017/de/dkrz/mistral`
- System was procured in two phases
  - 2015: Phase 1 with 31 PiB storage
  - 2016: Phase 2 with 21 PiB storage
- Other systems/services at DKRZ use Mistral's Lustre storage
- Lustre aspects
  - RobinHood for QoS and policy management
  - Lustre 2.5 Seagate edition (with patches from 2.7+)
  - University of Hamburg is IPCC for Lustre
    Researching file-system compression

I/O Architecture (Phase 1)

- 31 ClusterStor 9000 Scalable Storage Units (SSUs)
  - SSU: Active/Active failover server pair
- Single Object Storage Server (OSS)
  - 1 FDR uplink
  - GridRaid: (Object Storage Target (OST))
    - 41 HDDs, de-clustered RAID6 with 8+2(+2 spare blocks)
    - 1 SSD for the Log/Journal
  - 6 TByte disks
- 31 Extension units (JBODs)
  - Do not provide network connections
  - Storage by an extension is managed by the connected SSU
- Multiple metadata servers
  - Root MDS + 4 DNE MDS
  - Active/Active failover (DNEs, Root MDS with Mgmt)
  - DNE phase 1: Assign responsible MDS per directory

I/O Architecture (Phase 2)

- Adds another file system (Now two)
  - Both mounted on all compute nodes
- 34 ClusterStor L300 Scalable Storage Units (SSUs)
  - Uses a slightly different softwareF
- 34 Extension units (JBODs)
- Storage hardware
  - Seagate Enterprise Capacity V5 (8 TB) disks
- Multiple metadata servers
  - Root MDS + 7 DNE MDS

Parallel File System

Lustre 2.5 (Seagate edition, some backports from 2.7+)

Filesystem setup

- We have two file systems: /mnt/lustre0[1,2]
- Symlinks (for convenience): /work, /scratch, /home, ...
- For `mv`, each metadata server behaves like a file system

Assignment of MDTs to Directories

- In the current version, directories must be assigned to MDTs
  - /home/* on MDT0
  - /work/[projects] are distributed across MDT1-4
  - /scratch/[a,b,g,k,m,u] are distributed across MDT1-4
- Data transfer between MDTs is currently slow (mv becomes cp)
- We transfer projects to the phase 2 file system
  - New projects are only created on the phase 2 system

Peak Performance

Phase 1 + 2

- 65 SSUs · (2 OSS/SSU + 2 JBODs/SSU)
- 1 Infiniband FDR-14: 6 GiB/s ⇒ 780 GiB/s
- 1 ClusterStor9000 (CPU + 6 GBit SAS): 5.4 GiB/s
- L300 yield IB speed, still we consider 5.4 GiB/s ⇒ aggregated performance **704 GiB/s**
- Phase 2: `obd-filter` survey demonstrates that 480 GB/s and 580 GB/s can be delivered

Performance Results from Acceptance Tests

- Throughput in GB/s (% to peak) measured with IOR
  - Buffer size 2,000,000 (unaligned) on 42 OSS (Phase 1), 64 (P 2)
  - In the phase 2 testing, the RAID of at least one OSS is rebuilding

	Phase 1		Phase 2	
Type	Read	Write	Read	Write
POSIX, independent[1]	160 (70%)	157 (69%)	215 (62%)	290 (84%)
MPI-IO, shared	52 (23%)	41 (18%)	65 (19%)	122 (35%)
PNetCDF, shared	81 (36%)	38 (17%)	63 (18%)	66 (19%)
HDF5,shared	23 (10%)	24 (11%)	62 (18%)	68 (20%)
POSIX, single stream	1.1 (5%)	1.05 (5%)	0.98 (5%)	1.08 (5%)

- Metadata measured with Parabench / md-real-io pattern
  - Phase 1: 80 kOPs/s
    - 25 kOP/s for root MDS; 15 kOP/s for DNEs
  - Phase 2: 210 kOPs/s
    - 25 kOP/s for root MDS; 30-35 kOP/s for DNEs

[1] 1 stripe per file

Experience with Lustre

Performance issues

- Nearly full storage, performance drops considerably (expected)
  - File open/close degrades significantly
- Read latency of small files untolerable for interactive usage
  - We keep the software tree on Lustre
  - 10s to start certain apps (with hot cache!)
  - Mounting EXT4 volume on top of Lustre is faster (1s)
  - Evaluation of FUSE to cache data better
- DNE Phase 1: mv between directories trigers data movement instead of metadata movement
- Many tunables in kernel, application level
  - Defining number of stripes
  - Suboptimal data sieving in MPI-IO
- Test partition (1% capacity) to identify hw issues would be good

Experience with Lustre (2)

Reliability

- Had reliability issues with HA (firmware bug), caused bi-weekly reboot of the cluster
- Monthly verification of RAID integrity nice but impacts performance, now runs 2 weeks per month with low priority
- Experience frequent disconnects from clients to servers (100+ per day across the cluster)
- OST down for 1 hour+, shutdown of system not possible

Usability for Admins

- Load balancing between OSTs uses a homebrew solution
- Migrating data between the two file systems is painful
- RobinHood is active development
- Compatibility of Lustre clients sometimes suboptimal

Performance Monitoring

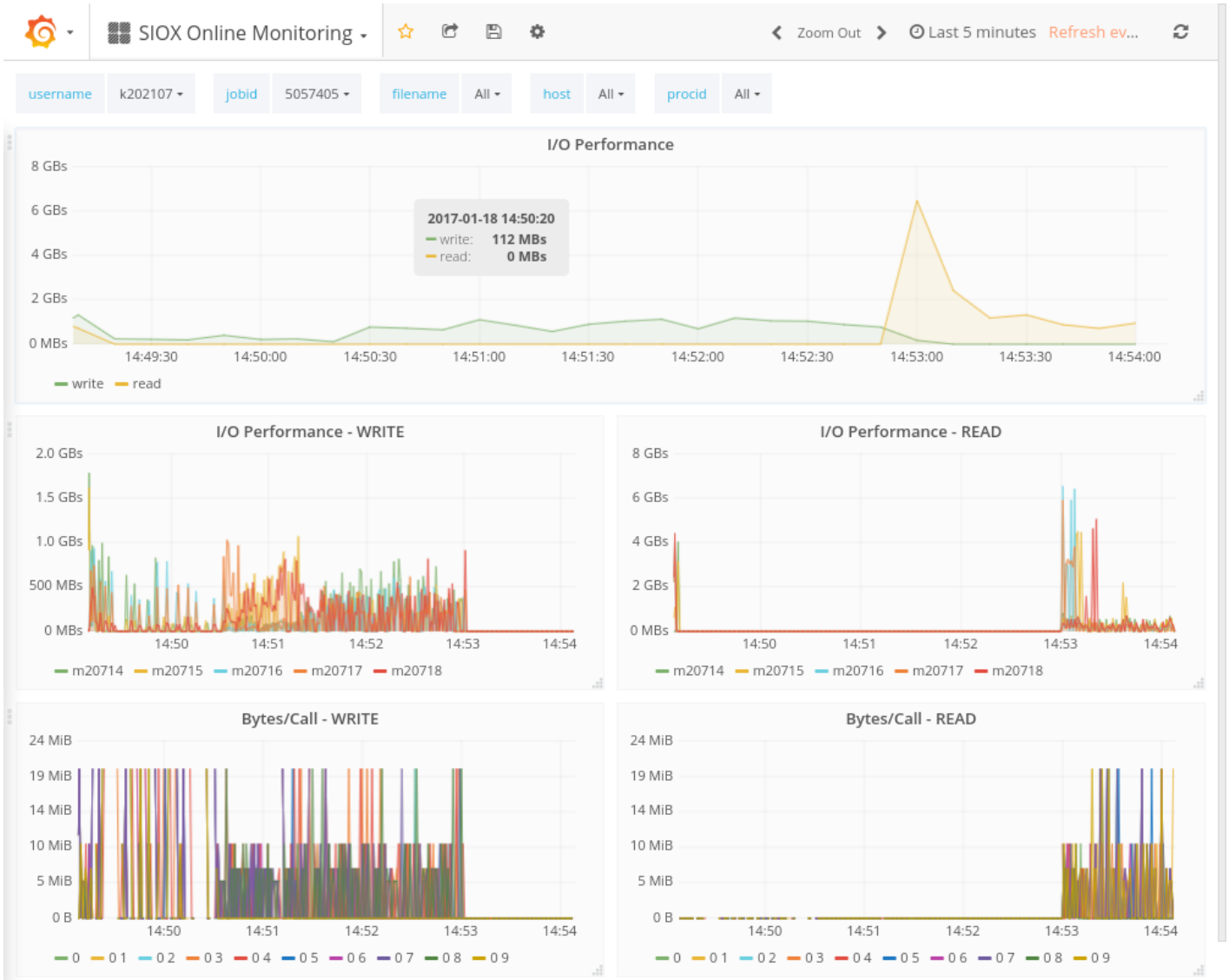
DKRZ's Approach

- We use Grafana for visualization
- Slurm job info is transferred to Lustre OSTs
  - Grafana allows visualization of OST performance per job
- Slurm extensions for client-side monitoring
  - Enable monitoring via: sbatch –monitoring
  - Client side monitoring of Lustre statistics (and cache efficiency)
- Monitoring of client I/O (and mmap) using FUSE with SIOX
  - Provides further information, traces of I/O possible
  - Online support
  - On demand mountable

Regression testing

- Daily regression testing with Jenkins using IOR
  - 20+ patterns, various stripe sizes, used APIs, workloads

Online Monitoring Example



Performance Regression with Full Servers

IOR runtime on 8 nodes for the Phase 1 file system