GWDG
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Performance of CephFS for HPC and AI

and the new HPC storage concepts at GWDG

Sebastian Krey Patrick Höhn

27.02.2025

Outline

1 Current HPC Storagesystem at GWDG

2 HPC storage concept

3 Experiences with current storage

4 The new storage systems

5 MCSE

Storage Systems

- WORK MDC: DDN ExaScaler 5 EoL 08/24
  - Metadata SFA7700X
  - 8 PiB HDD 2x ES14KX
  - 113 TiB NVME 2x SFA200NV
- WORK MDC new: 7 Celestica SC6100 1.3 PiB NVME (from 03/25)
- WORK RZGÖ: DDN ExaScaler 6 510 TiB NVME 2x ES400NVX
- HOME/SW/WORK KISSKI: VAST Data 1.1PiB NVME (3x dBox, 3x cBox)
- WORK SCC: 2.2 PiB BeeGFS based on DDN SFA7990 block storage
- HOME SCC: 3 PiB Quantum StorNext
- HSM/Tape: Quantum StorNext HSM 60+ PiB

Current storage concept

- Different user groups have different storage systems available
- The same path (e.g. `/scratch`) can point to filesystems with different characteristics.
- Not all storage systems are available on all nodes
- Different concepts for data sharing depending on source of project/user (compute projects, functional accounts, etc.)
- Unified operation requires same storage access for all nodes and currently not possible accross all systems
- Users of Tier 3 system have their campus home as home directory

New unified storage concept for NHR/SCC/KISSKI

- Replace HDD based WORK storage with central Ceph instance
- Compute island specific high performance storage, all flash (Lustre, VAST or BeeGFS, DAOS maybe a candidate in the future)
- Unify HOME/SW to central HPC home storage
- HPC S3 object storage for “Cloud” workloads and easy data ingest/export with central S3 storage of infrastructure group and external parties
- Access to campus home directory (StorNext) only via data mover nodes
- Semantic storage: Assignment of storage backend based on project requirements, transparent access via symlinks.
- Directory quotas, whenever possible

Storage assignment

- Based on project application space and filesystem type will be assigned
- Every user gets home directories for their project specific user accounts
- Every project gets their volume storage in the central coldstorage
- Every project gets archive storage based on requirements
- In RZGÖ assingment of high performance storage based on I/O requirements (Lustre or VAST depending on read/write mix)
- Open question: Management of campaign storage
  - Admin assignment or self management by user/project → Workspaces

Problems with current storage

Lustre: long failover times and crashes happen quite often, enterprise support usually way behind open source version regarding kernel support, software and hardware support linked

BeeGFS: lacks some features like project/directory quota, performance scaling in larger NVME setups

GPFS: expensive, strict kernel version requirements, no directory quota, metadata handling on client can be advantage and disadvantage, limited fabric support, licensing and features depends on used hardware

StorNext: architecture outdated (focus on SAN, single MDS), slow bugfixing and updates for kernel support, current licensing model expensive, missing a lot of modern features, software and hardware support linked

Strengths of the different storage technologie

VAST: Extreme high availability, NFS everywhere useable, high read speed, consistent low latencies

Lustre: Extreme high performance possible, user configurable striping

BeeGFS: Very easy to setup and manage, good performance

GPFS: Can a lot of stuff, good performance possible

Ceph: Capacity scaling, very low EUR/TB, full independancy from hardware vendors, properly setup: good performance

Storage Systems: New Homestorage

- Unified home storage for all user groups
- Expansion of existing VAST storage
- 1.1 PiB all flash total capacity
- Mounted via NFS on all compute nodes
- Will also provide the central software installation
- Strict volume quota, relaxed inode quota
- Daily snapshots and offsite backup

Storage Systems: New High Performance storage

- Expansion of WORK RZGÖ (Lustre) to 510 TiB (SSD replacement)
- New Lustre based filesystem for WORK MDC (1.3 PiB)
- Using extra capacity of VAST for read intensive AI workloads
- Usage limited to specific compute island to ensure high performance
- Strict volume and inode quota
- All flash filesystems to allow best performance in all workload types
- Smaller HPC hosting: BeeGFS for easy setup and maintenance

Storage Systems: New Coldstorage

Hardware:

- 53 Servers, 21 PB HDD, 3.5 PB NVME
- HDD Cluster with 45 Servers:
  - 24x 20TB HDD, 4x 7.68 NVME
  - 2x24 Core Sapphire Rapids CPUs, 512 GB memory
  - 2x25G Ethernet
- NVME Cluster with 8 Servers
  - 20x 15.36TB NVME
  - 2x32 Core Milan CPUs, 512GB memory
  - 100G Ethernet
- HDD cluster capacity optimized → Erasure Coding
- NVME cluster performance optimized → Replication
- Enterprise support from “Clyso”

Ceph for HPC?

Common opinion:

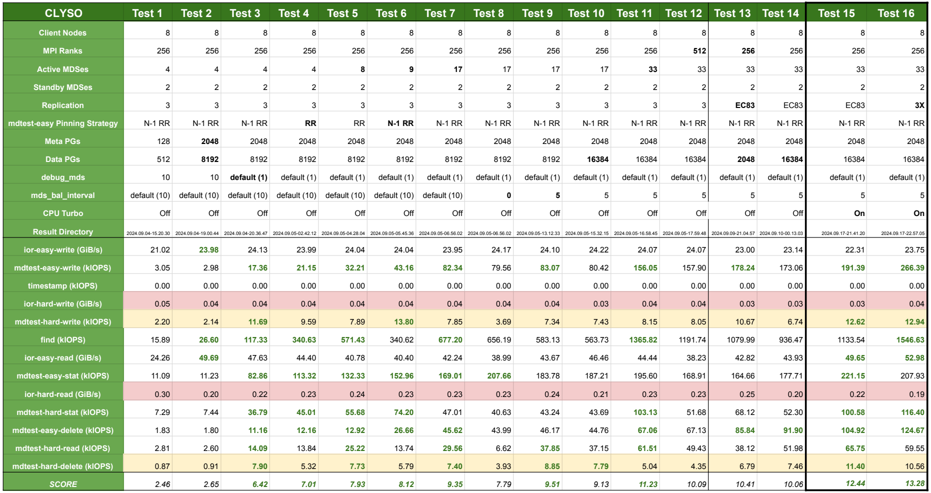
- Are you insane?
- Ceph is slow, complex, unreliable,. . .
- Only TCP connections

On closer look:

- Ceph is reliable standard in cloud environments
- Some institutes use it successfully in HPC (e.g. CERN, IZUM)
- Ceph allows complete hardware vendor independence
- Hardware migrations in live operation, without user interaction
- Recent performance improvements show respectable performance (work from Clyso and Croit)
- With enough CPU cores and memory 75-80% network saturation
- Enough MDS containers achieves very good metadata performance scaling

Ceph IO500 Performance

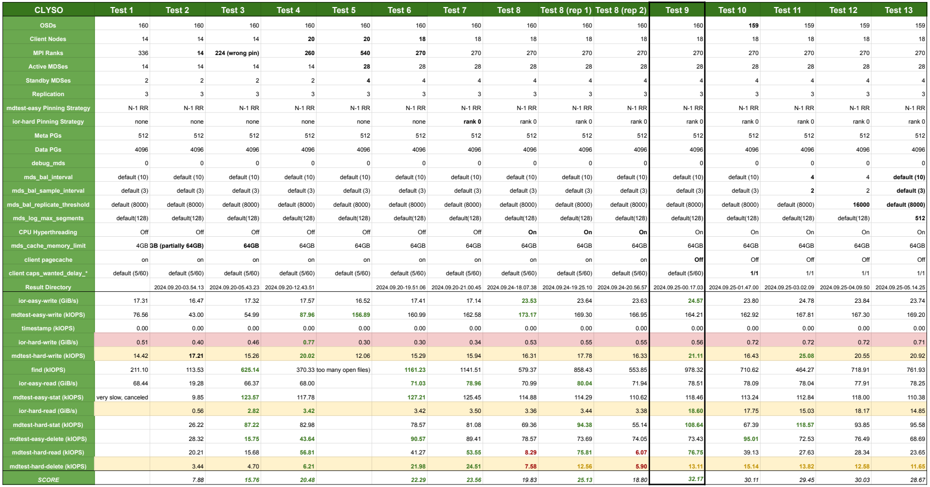
First benchmark results:



CLYSO	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Test 11	Test 12	Test 13	Test 14	Test 15	Test 16
Client Nodes	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
MPI Ranks	256	256	256	256	256	256	256	256	256	256	256	**512**	**256**	256	256	256
Active MDSes	4	4	4	4	**8**	**9**	**17**	17	17	17	**33**	33	33	33	33	33
Standby MDSes	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Replication	3	3	3	3	3	3	3	3	3	3	3	3	**EC83**	EC83	EC83	**3X**
mdtest-easy Pinning Strategy	N-1 RR	N-1 RR	N-1 RR	**RR**	RR	**N-1 RR**	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR
Meta PGs	128	**2048**	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048
Data PGs	512	**8192**	8192	8192	8192	8192	8192	8192	8192	**16384**	16384	16384	**2048**	**16384**	16384	16384
debug_mds	10	10	**default (1)**	default (1)	default (1)	default (1)	default (1)	default (1)	default (1)	default (1)	default (1)	default (1)	default (1)	default (1)	default (1)	default (1)
mds_bal_interval	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	**0**	**5**	5	5	5	5	5	5	5
CPU Turbo	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	**On**	**On**
Result Directory	2024.09.04-15.20.30	2024.09.04-19.00.44	2024.09.04-20.36.47	2024.09.05-02.42.12	2024.09.05-04.28.04	2024.09.05-05.45.36	2024.09.05-06.56.02	2024.09.05-06.56.02	2024.09.05-13.12.33	2024.09.05-15.32.15	2024.09.05-16.58.45	2024.09.05-17.59.48	2024.09.09-21.04.57	2024.09.10-00.13.03	2024.09.17-21.41.20	2024.09.17-22.57.05
ior-easy-write (GiB/s)	21.02	23.98	24.13	23.99	24.04	24.04	23.95	24.17	24.10	24.22	24.07	24.07	23.00	23.14	22.31	23.75
mdtest-easy-write (kIOPS)	3.05	2.98	17.36	21.15	32.21	43.16	82.34	79.56	83.07	80.42	156.05	157.90	178.24	173.06	191.39	266.39
timestamp (kIOPS)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ior-hard-write (GiB/s)	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.03	0.04	0.04	0.03	0.03	0.03	0.04
mdtest-hard-write (kIOPS)	2.20	2.14	11.69	9.59	7.89	13.80	7.85	3.69	7.34	7.43	8.15	8.05	10.67	6.74	12.62	12.94
find (kIOPS)	15.89	26.60	117.33	340.63	571.43	340.62	677.20	656.19	583.13	563.73	1365.82	1191.74	1079.99	936.47	1133.54	1546.63
ior-easy-read (GiB/s)	24.26	49.69	47.63	44.40	40.78	40.40	42.24	38.99	43.67	46.46	44.44	38.23	42.82	43.93	49.65	52.98
mdtest-easy-stat (kIOPS)	11.09	11.23	82.86	113.32	132.33	152.96	169.01	207.66	183.78	187.21	195.60	168.91	164.66	177.71	221.15	207.93
ior-hard-read (GiB/s)	0.30	0.20	0.22	0.23	0.24	0.23	0.23	0.23	0.24	0.21	0.23	0.23	0.25	0.20	0.22	0.19
mdtest-hard-stat (kIOPS)	7.29	7.44	36.79	45.01	55.68	74.20	47.01	40.63	43.24	43.69	103.13	51.68	68.12	52.30	100.58	116.40
mdtest-easy-delete (kIOPS)	1.83	1.80	11.16	12.16	12.92	26.66	45.62	43.99	46.17	44.76	67.06	67.13	85.84	91.90	104.92	124.67
mdtest-hard-read (kIOPS)	2.81	2.60	14.09	13.84	25.22	13.74	29.56	6.62	37.85	37.15	61.51	49.43	38.12	51.98	65.75	59.55
mdtest-hard-delete (kIOPS)	0.87	0.91	7.90	5.32	7.73	5.79	7.40	3.93	8.85	7.79	5.04	4.35	6.79	7.46	11.40	10.56
SCORE	*2.46*	*2.65*	*6.42*	*7.01*	*7.93*	*8.12*	*9.35*	*7.79*	*9.51*	*9.13*	*11.23*	*10.09*	*10.41*	*10.06*	*12.44*	*13.28*

Ceph IO500 Performance

First benchmark results:



CLYSO	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 8 (rep 1)	Test 8 (rep 2)	Test 9	Test 10	Test 11	Test 12	Test 13
OSDs	160	160	160	160	160	160	160	160	160	160	160	**159**	159	159	159
Client Nodes	14	14	14	**20**	**20**	**18**	18	18	18	18	18	18	18	18	18
MPI Ranks	336	**14**	**224 (wrong pin)**	**260**	**540**	**270**	270	270	270	270	270	270	270	270	270
Active MDSes	14	14	14	14	**28**	28	28	28	28	28	28	28	28	28	28
Standby MDSes	2	2	2	2	**4**	4	4	4	4	4	4	4	4	4	4
Replication	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
mdtest-easy Pinning Strategy	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR	N-1 RR
ior-hard Pinning Strategy	none	none	none	none	none	none	**rank 0**	rank 0	rank 0	rank 0	rank 0	rank 0	rank 0	rank 0	rank 0
Meta PGs	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512
Data PGs	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096
debug_mds	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mds_bal_interval	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	default (10)	**4**	4	**default (10)**
mds_bal_sample_interval	default (3)	default (3)	default (3)	default (3)	default (3)	default (3)	default (3)	default (3)	default (3)	default (3)	default (3)	default (3)	**2**	2	**default (3)**
mds_bal_replicate_threshold	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	default (8000)	**16000**	**default (8000)**
mds_log_max_segments	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	default(128)	**512**
CPU Hyperthreading	Off	Off	Off	Off	Off	Off	Off	**On**	**On**	**On**	On	On	On	On	On
mds_cache_memory_limit	4GB	**GB (partially 64GB)**	**64GB**	64GB	64GB	64GB	64GB	64GB	64GB	64GB	64GB	64GB	64GB	64GB	64GB
client pagecache	on	on	on	on	on	on	on	on	on	on	**Off**	Off	Off	Off	Off
client caps_wanted_delay_*	default (5/60)	default (5/60)	default (5/60)	default (5/60)	default (5/60)	default (5/60)	default (5/60)	default (5/60)	default (5/60)	default (5/60)	default (5/60)	**1/1**	1/1	1/1	1/1
Result Directory		2024.09.20-03.54.13	2024.09.20-05.43.23	2024.09.20-12.43.51		2024.09.20-19.51.06	2024.09.20-21.00.45	2024.09.24-18.07.38	2024.09.24-19.25.10	2024.09.24-20.56.57	2024.09.25-00.17.03	2024.09.25-01.47.00	2024.09.25-03.02.09	2024.09.25-04.09.50	2024.09.25-05.14.25
ior-easy-write (GiB/s)	17.31	16.47	17.32	17.57	16.52	17.41	17.14	**23.53**	23.64	23.63	**24.57**	23.80	24.78	23.84	23.74
mdtest-easy-write (kIOPS)	76.56	43.00	54.99	**87.96**	**156.89**	160.99	162.58	**173.17**	169.30	166.95	164.21	162.92	167.81	167.30	169.20
timestamp (kIOPS)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ior-hard-write (GiB/s)	0.51	0.40	0.46	**0.77**	0.30	0.30	0.34	0.53	0.55	0.55	0.56	0.72	0.72	0.72	0.71
mdtest-hard-write (kIOPS)	14.42	**17.21**	15.26	**20.02**	12.06	15.29	15.94	16.31	17.78	16.33	**21.11**	16.43	**25.08**	20.55	20.92
find (kIOPS)	211.10	113.53	**625.14**	370.33	too many open files)	**1161.23**	1141.51	579.37	858.43	553.85	978.32	710.62	464.27	718.91	761.93
ior-easy-read (GiB/s)	68.44	19.28	66.37	68.00		**71.03**	**78.96**	70.99	**80.04**	71.94	78.51	78.09	78.04	77.91	78.25
mdtest-easy-stat (kIOPS)	very slow, canceled	9.85	**123.57**	117.78		**127.21**	125.45	114.88	114.29	110.62	118.46	113.24	112.84	118.00	110.38
ior-hard-read (GiB/s)		0.56	**2.82**	**3.42**		3.42	3.50	3.36	3.44	3.38	**18.60**	17.75	15.03	18.17	14.85
mdtest-hard-stat (kIOPS)		26.22	**87.22**	82.98		78.57	81.08	69.36	**94.38**	55.14	**108.64**	67.39	**118.57**	93.85	95.58
mdtest-easy-delete (kIOPS)		28.32	**15.75**	**43.64**		**90.57**	89.41	78.57	73.69	74.05	73.43	**95.01**	72.53	76.49	68.69
mdtest-hard-read (kIOPS)		20.21	15.68	**56.81**		41.27	**53.55**	**8.29**	**75.81**	**6.07**	**76.75**	39.13	27.63	28.34	23.65
mdtest-hard-delete (kIOPS)		3.44	4.70	**6.21**		**21.98**	**24.51**	**7.58**	**12.56**	**5.90**	**13.11**	**15.14**	**13.82**	**12.58**	**11.65**
SCORE		*7.88*	***15.76***	***20.48***		***22.29***	***23.56***	*19.83*	***25.13***	*18.80*	***32.17***	*30.11*	*29.45*	*30.03*	*28.67*

Comparison with HDD Lustre

	Lustre	CephFS
ior-easy-read	26-60	38-44
ior-easy-write	50-56	23-24
mdtest-easy-write	120k	160k
mdtest-easy-stat	300k	200k
mdtest-hard-write	20k	12k
mdtest-hard-read	27k	40-60k

CephFS impressions

- Performance way better than expected
- For HDD system compareable with our Lustre from 2018
- Metadata servers also look like Lustre <2.10 (high single core performance needed)
- Best metadata performance with a load distribution like Lustre DNE 1
- Automatic sharding can speed up very large directories, but dynamic process (unlike Lustre DNE 2), so high performance variation at beginning
- Option to change storage layout per directory can help for small file workloads

MCSE (Memory Centric Storage for Exascale) Projektziele

- Untersuchung der Semantiken bestehender I/O API’s
- Entwicklung einer API (IOVerbs), um paralleles I/O elementar ausdrücken zu können sowie I/O Semantiken explizit zu transportieren.

⇒ Einheitliche Schnittstelle für Arbeitsspeicher und nichtflüchtigen Speicher

- Verschiedene Klassen von Speichermedien flexibel in HPC-Workflows (Kampagnen) nutzen
- Entwicklung von Memory-Centric Storage System (MCS2)
- Angebot eines einfachen Migrationspfads von bestehenden Anwendungen.

MCS2 und Workflow System

MCS2

- Gleichberechtigter C++ API und CLI.
- Beliebig konfigurierte Storages gemeinsam nutzbar.
- Clients agnostisch gegenüber Storage Konfigurationen.
- Notwendiger Datentransport direkt zwischen den beteiligten Storages.
- Bereitstellung von Speicher mit Ablaufdatum.

Workflow System

- Integration von MCS2 in SnakeMake.
- Kette der IO-Operationen bestimmt Ausführungsreihenfolge.
- Integration
  - durch CLI-Befehle von MCS2 in SLURM Batch Skripts.
  - in allgemeinen resources Abschnitt der Regeldefinition.
  - innerhalb default-resources des SLURM-Plugins.
  - als separates Storage-Plugin.

Summary

- CephFS is a viable approach for providing HPC storage
- Frame contracts for standard servers can be used
- Performance for HDD workloads compareable to other storage
- NVME performance better than expected, sufficient for a lot of workloads
- Setup complexity with `cephadm` managed containers under control
- Good professional support available
- S3 interesting for migrating cloud workloads to HPC clusters and data transfer
- MCS2 will allow easier usage of different storage sytems and migration of IO workflows to different clusters