HPS

Hendrik Nolte, Freja Nordsiek, Sebastian Krey, Julian Kunkel

Understanding Storage Performance using Benchmarking Experiences at GWDG

The GWDG: HPC storage environment

- Work storage for scratch and project storage space
  - Emmy: DDN Lustre (8.5 PiB HDD, 130 TiB NVME)
    - https://www.top500.org/system/179883/
  - Grete: DDN Lustre (130 TiB NVME and LNET routing to Emmy storage)
    - https://www.top500.org/system/180092/
  - CARO: DDN Lustre (8 PiB HDD, 200 TiB SSD)
    - https://www.top500.org/system/180038/
  - KISSKI: VAST Data (600 TiB, 1 Lightstream dBox, 1 cBox IB, 1 cBox 100GbE)
  - SCC: BeeGFS based on DDN blockstorage (2.1 PiB HDD, 100 TiB SSD)
- Home storage
  - Tier 2: DDN GridScaler (GPFS/Storage Scale via NFS, 350TiB)
  - Tier 3: GWDG central UNIX home directory (Quantum StorNext, DLC and NFS)
- Archive
  - Tier 2: Quantum Tape Library with StorNext HSM (7PB gross)
  - Tier 3: GWDG central StorNext HSM
- Upcoming: Large central cold storage procurement

Why Doing Storage Benchmarks

- Verification of vendor performance specifications
- Understanding the conditions for achieving good performance
- Understanding the reasons for bad performance
- Diagnosing bottlenecks and technical problems
- Testing new configurations
- Assess different systems for new procurement

→ Need a quick convergence of benchmark settings

Benchmarks - IO500

- Core benchmarks IOR/MDTest/MDWorkbench support GPUDirect
  - Normally, IO is done between Client NIC + (host) memory
  - GPUDirect: IO is done between Client NIC + GPU memory - skipping host mem
- We can choose if data buffers and patterns are created/verified on GPU/CPU
- Extra flag: allocateBufferOnGPU=MODE

Mode	Buffer	Creation, Verification (if enabled)	GPUDirect
0	malloc()	CPU	No
1	cudaMallocManaged()	CPU	Optional
2	cudaMallocManaged()	GPU	Optional
3	cudaMalloc()	GPU	Mandatory

- To enable GPUDirect: –gpuDirect
- Requires: POSIX odirect flag
- Limitations: Verification is currently supported only for timestamp pattern

Preliminary Results

Task / Mode	0	1	2	2-GPUD	3-GPUD
ior-easy-write [GiB/s]	6.3	7.5	7.0	6.1	5.6
ior-rnd4K-write [GiB/s]	0.2	0.2	0.19	0.02	0.02
mdtest-easy-write [kIOPS]	11.7	11.5	11.5	8.4	8.3
ior-rnd1MB-write [GiB/s]	1.2	0.9	1.2	5.2	3.8
mdworkbench-create	11.0	11.0	11.0	3.4	3.3
find-easy [kIOPS]	2635.5	2219.4	2501.2	2241.7	2433.7
ior-hard-write [GiB/s]	0.7	0.4	0.4	0.1	0.1
mdtest-hard-write [kIOPS]	2.5	2.8	2.8	2.5	2.3
find [kIOPS]	1577.8	1543.6	1473.7	2713.4	2624.0
ior-rnd4K-read [GiB/s]	2.4	0.1	0.2	0.03	0.03
ior-rnd1MB-read [GiB/s]	26.9	2.8	3.3	5.2	4.2
find-hard [kIOPS]	1364.6	1655.5	1398.9	1342.5	1201.2
mdworkbench-bench [kIOPS]	18.6	18.3	3.1	8.4	2.9
concurrent [score]	6.5	6.3	3.6	7.7	4.9
ior-easy-read [GiB/s]	5.8	6.1	3.6	6.2	6.1
mdtest-easy-stat [kIOPS]	28.8	28.7	29.6	207.8	200.0
ior-hard-read [GiB/s]	3.0	2.2	0.24	0.3	0.3
mdtest-hard-stat [kIOPS]	49.5	49.7	46.4	194.0	190.8
mdworkbench-find-delete [kIOPS]	19.9	19.9	20.8	19.9	20.1
mdtest-easy-delete [kIOPS]	21.8	22.3	19.7	22.0	20.0
mdtest-hard-read [kIOPS]	14.4	14.4	4.9	5.0	5.0
mdtest-hard-delete [kIOPS]	5.0	4.9	4.9	5.1	4.7
Score Bandwidth [GiB/s]	2.9	2.5	1.2	1.0	1.0
Score IOPS [kIOPS]	23.8	24.1	20.4	32.6	31.2
ScoreX Bandwidth [GiB/s]	2.4	1.1	0.6	0.6	0.5
ScoreX IOPS [kIOPS]	47.6	48.0	37.1	54.2	48.0

- Number shows the mode
- GPUDirect on/off
- Used a single node on Grete
  - 9 processes
  - 3 GPUs
- Numbers are irrelevant
  $\rightarrow$ Just want to show it works
  $\rightarrow$ And how it looks like
- Shows volatility to file count
  slowdown due md create good perf
- Find/Easy hard consistent results, not find.

Example: Multi-Rail Storage Nodes

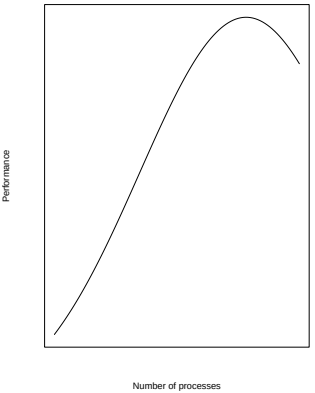
- Advantages
  - Can have more disks per node before hitting network bandwidth limit
  - Provides fallback interfaces
- Difficulties
  - Answers need to come from the same rail from which the request was received
  - Must have one server per rail for clients that don't load-balance
- Early results
  - For elbencho big read tests on 80 nodes with 32 threads and 1 x 100 Gib/s OPA on non-mirroring BeeGFS with 8 storage nodes with 2 x 100 GiB OPA:

servers	read (Gib/s)	write (Gib/s)
1	72.2	41.3
2	78.9	55.6

  - Need more CPU cores and RAM per storage node
  - Risk too many RDMA interfaces on the same NUMA domain,
    - Manual pinning of interrupts to cores

Include expert knowledge

- Storage experts can give good baseline configurations based on specific criteria of the storage system
- Number of drives and type of drive, optimal number of IO processes $n \times t = k \times d$
- Metadata ($k = 16+$), HDD ($k = 1 - 2$) as well as SSD ($k = 4 - 8$) based object storage targets have different known performance characteristics
- Performance of network interconnect
- Theoretical limit of storage controller hardware
- Performance commitment of the vendor



benchscale

How to run efficiently specific IO500 benchmark configurations?

`benchscale` is a small tool written in Rust to execute specific benchmarks of the IO500 suite in defined configurations (client nodes and number of processes per node) and output the relevant performance metrics.

Currently still in early development, but major progress in the next weeks expected.

Public release of source code pending.

Inclusion of elbencho benchmark tool, for removing MPI requirement in discussion.

First results

- Explore slope and scaling limit for the IOR Easy sub-benchmark
- Start with a single node and one process
- Estimate number of required processes based on first result and capabilities of network interconnect
- First large-scale run based on number of drives
- High performance run based on slope between run 2 and 3

SSD Lustre with 46 NVME SSDs

Nodes/PPN	1/1	1/16	12/16	16/16
Max Bandwidth	0.83	4.47	29.35	33.71

Peak performance of 37.14 GiB/s with 32 nodes and 16 processes, so 91% of peak achieved.

HDD Lustre with 1000 HDDs

Nodes/PPN	1/1	1/18	56/18	78/18
Max Bandwidth	0.64	6.49	51.8	51.85

Peak performance of 52.47 GiB/s with 64 nodes and 32 processes, so 99% of peak achieved.

IO Weather Map

- GWDG uses monitoring and benchmarks to assess compute/storage
  - Grafana for monitoring
  - Are exploring how to export Grafana data to individual users
- Regression tests track performance change over time (and system change)
- We are about to deploy benchmarks to report I/O-weather to the users
- Utilized benchmarks
  - IO500 benchmark
  - elbencho
  - Minio/warp