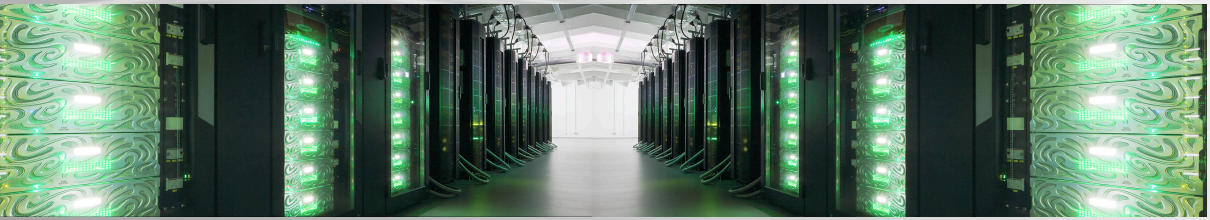
GWDG
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

NHR-Nord@Göttingen

Holistic HPC I/O

Benchmarking and IO500

Mohammad Hossein Biniaz, Kevin Lüdemann



12.06.2024

NHR Summer School 2024

Table of contents

Objectives

- What benchmarking is, basically, the theoretical knowledge behind it.
- Why benchmarking is done, i.e., necessity of benchmarks.
- How benchmarking is done, i.e., ways of doing benchmarking.
- Using the IO500

What is Benchmarking?

Theoretical introduction of benchmarking.

Benchmark (noun)

- A standard or point of reference against which things can be compared.

Benchmarking (verb)

- Process of comparing with a previously defined standards (benchmarks).

https://languages.oup.com/google-dictionary-en

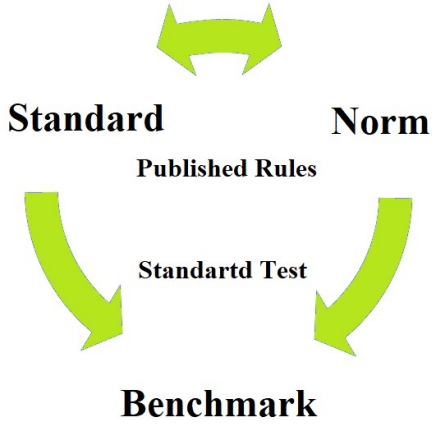
What is Benchmarking?

Standards!

- Need Standards for Benchmarking?

Logically,

- Standards are published rules, and
- Prerequisites for any standards are,
  - Should be technically mature, and
  - Should have benefits for the users.



https://de.wikipedia.org/wiki/Standard

Be Careful: Certified Standards Does Not Certify Benchmarks

Standard

- It is a published specification ratified by some organization.
- e.g., ISO 9001 - is an international standard for quality management.

Benchmark

- It is a test to evaluate your system's performance,
- It is often established by general organizational acceptance.
- e.g., IO 500 Benchmark - is a comparison against standards.

Images: https://www.iso.org/modules/isoorg-template/img/iso/iso-logo-print.gif, https://www.vi4io.org/io500/start, https://www.top500.org/news/chinas-tianhe-2-supercomputer-retains-top-spot-on-43rd-edition-of-the-top500-list/

Why Benchmarking is Done?

HPC Benchmarking

- Benchmark measures system behavior, so
- Whenever there is question about performance, answer is benchmarking.

Moreover

- Benchmarking measures the relative performance either by,
  - Changing the in/out parameters, or
  - On scaling the system.
- Measured by running a number of standard tests and programs. For e.g.,
  - Running a computer program (micro benchmarking),
  - A collection of programs (macro benchmarking),
  - Other operations (overall benchmarking).

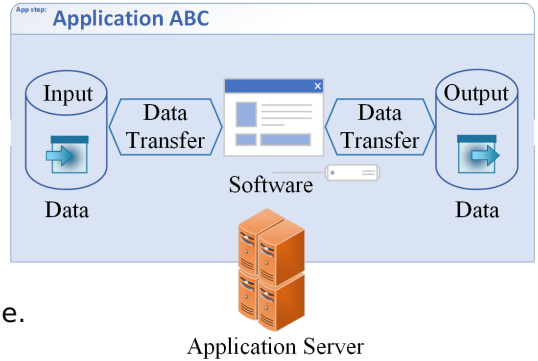
.

Fleming and Wallace, “How Not to Lie with Statistics”, 1986

What is an application and how it is benchmarked?

Application means: (System & Workload)

- The configured system,
- Software running on it, and
- Input/Output data required by it.



So to benchmark need

- To calibrate the whole system.
- To allocate proper resources.
- To make sure there is ideal wait time.

How to Benchmark: What are the Benchmarking Key Metrics

Key Metrics

- Micro Benchmarking - Baseline performance
  - Measures benchmarking performance improvement against unit of node.
- Macro Benchmarking - Scaling
  - Measures how the performance changes with the number of nodes/cores.
- Overall Benchmarking - Performance
  - It is a measure of rate of how well an application is running.
- Other Measures
  - Timing
    - It is a measure of full or partial run-time of an application. Mainly wall clock.
  - Parallel efficiency
    - The ratio of measured scaling to the perfect scaling.

Let us Discussion on Macro Benchmarking - Scaling

- Scalability in case of Speedup for,
  - Hardware: is the ability to handle more workload by scaling compute power.
  - Software: is the parallelization efficiency, given by
    - The ratio of the actual speedup and the ideal speedup for a number of processors.

$$Speedup = t(1)/t(N) \longrightarrow t : time; N : no..of..processors. \quad (1)$$

Discussion on Macro Benchmarking - Scaling

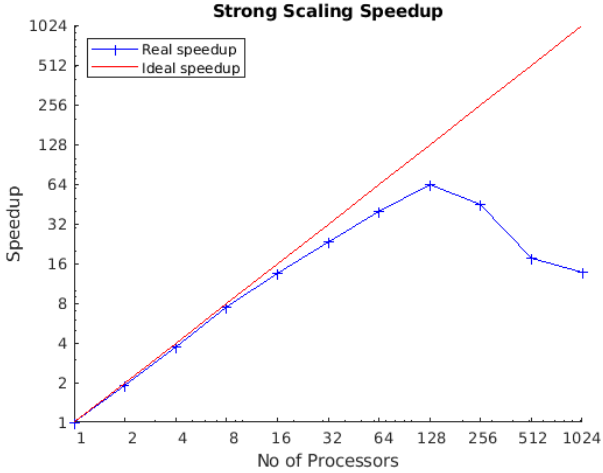
- Applications' speedup scaling test is done in two ways:
  - Strong Scaling
    - Number of processors is increased while the problem size remains constant.
    - e.g., Amdahl's law (1967),

$$Speedup = 1/(s + p/N) \longrightarrow s : serial; p : parallel \tag{2}$$

  - Week Scaling.
    - Both the number of processors and the problem size are increased.
    - e.g., Gustafson's law (1988),

$$Speedup = s + p * N \longrightarrow s : serial; p : parallel \tag{3}$$

An Example of Strong Scaling



Scaling - HPC Wiki

Guidelines on Scaling Benchmark

1. Measure using job sizes that span orders of magnitude
2. Use wall-clock time units or equivalent
3. Measure multiple independent runs per job size
4. Various factors must be considered when using more than one node:
   a) Interconnect speed and latency
   b) Max memory per node
   c) processors per node
   d) max processors (nodes)
   e) system variables and restrictions (e.g. stack size)
5. Also, if possible measure using different systems and factors.
6. Use a problem state that best matches the intended production runs.

In Summary

Overall Guidelines:

- Be alert and vigilante to the details,
- Think critically and include all the details,
- Use proper measures and charts to present,
- Adapt and address the changes properly,
- Attempt repetitively and continuously.

Image: https://thefruitfultoolbox.com/dos-donts-disc/

References

Fleming, Philip J. and John J. Wallace. “How Not to Lie with Statistics: The Correct Way to Summarize Benchmark Results”. In: *Commun. ACM* 29.3 (Mar. 1986), pp. 218–221. ISSN: 0001-0782. DOI: 10.1145/5666.5673.

Scaling - HPC Wiki. URL: https://hpc-wiki.info/hpc/Scaling (visited on 04/18/2023).

How Can Benchmarks Help to Analyze I/O?

- Benefits of benchmarks
  - Can use simple/understandable sequence of operations
    - Ease comparison with theoretic values (that requires understandable metrics)
  - May use a pattern like a realistic workloads
    - Provides performance estimates or bounds for workloads!
  - Sometimes only possibility to understand hardware capabilities
    - Because the theoretic analysis may be infeasible
- Benefits of benchmarks vs. applications
  - Are easier to code/understand/setup/run than applications
  - Come with less restrictive "license" limitations
- Flexible testing (strategies)
  - Single-shot: e.g., acceptance test
  - Periodically: regression tests

Benchmarks

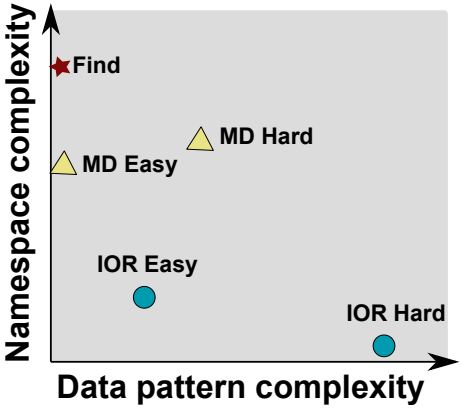
- Benchmarks measure system behavior and implement (simple) well-known behavior
- Many I/O benchmarks exist covering various aspects
  - APIs used
  - Data access pattern
  - Memory access pattern
  - Parallelism and concurrency
- Let's talk about the IO-500 benchmark suite; it is
  - **Representative**: for optimized and naive workloads
  - **Inclusive**: cover various storage technology and non-POSIX APIs
  - **Trustworthy**: representative results and prevent cheating
  - **Cheap**: easy to run and short benchmarking time (in the order of minutes)
  - Favors a single metric to simplify the comparison across dimensions

Goals of the IO-500 Benchmarking Effort

- Bound performance expectations for realistic workloads
- Track storage system characteristics behavior over the years
  - Foster understanding of storage performance development
  - Support to identify potent architectures for certain workloads
- Document and share best practices
  - Tuning of the system is encouraged
  - Submitters must submit detailed run parameters
- Support procurements, administrators and users

https://io500.org

Covered Access Patterns



- IOR-easy: large seq on file(s)
- IOR-hard: small random shared file
- MD-easy: mdtest, per rank dir, empty files
- MD-hard: mdtest, shared dir, 3900 byte
- find: query and filter files based on name and creation time
- Executing concurrent patterns not covered (another dimension)

Predictability and Latency Matters

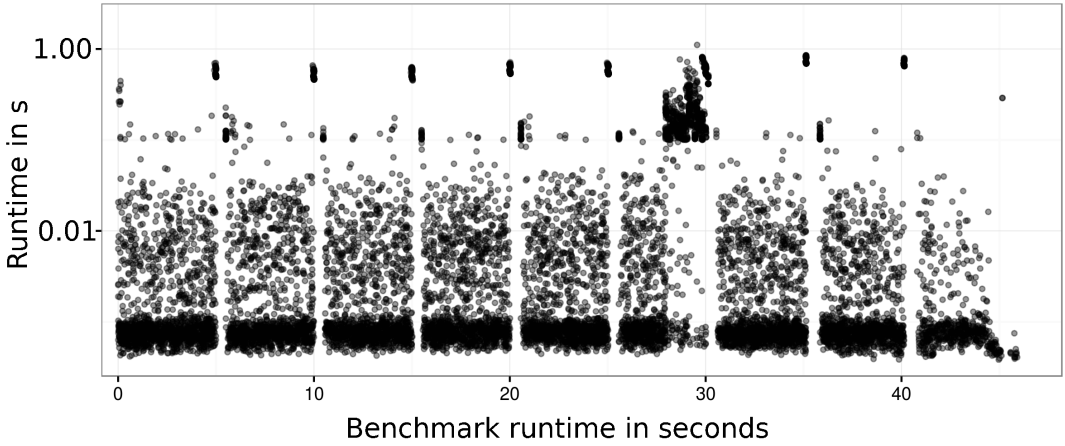
Performance Predictability

- How long does an I/O / metadata operation take?
- Important to predict runtime
- Important for bulk-synchronous parallel applications
  - The slowest straggler defines the performance

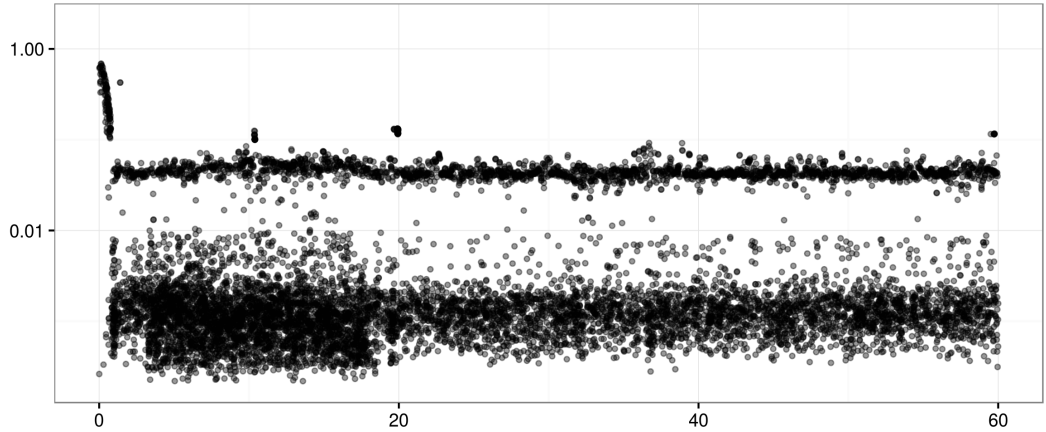
Measurement

- In the following, we plot the timelines of metadata create operations
  - Sparse plot with randomly selected measurements
  - Every point above 0.1s is added
- All results obtained on 10 Nodes using MD-Workbench
  `https://github.com/JulianKunkel/md-workbench`
  - Options: 10 PPN, D=1, I=2000, P=10k, precreation phase

Latencies: Lustre / Mistral at DKRZ



Latencies: GPFS / Cooley at ALCF



Importance of Choosing the Right Mean Value

- We must repeat a benchmark run to obtain trustworthy data
  - Reduce impact of random errors due to background activity
- How do we weight input when repeating a benchmark run?

Tuning for improving the Geom-Mean value

Description	Input (11 values)	**Geom**	Arithmetic	Harmonic
Balanced system	10 . . . 10 10 **10**	10	10	10
One slow bench	10 . . . 10 10 **1**	8.1	9.2	5.5
Tuning worst 2x	10 . . . 10 10 **2**	8.6	9.3	7.3
Tuning good 2x	10 . . . 10 **20 1**	8.6	10.1	5.6
Tuning good 100x	10 . . . 10 **100 1**	10	17.4	5.8

- Avoid arithmetic mean
- May use box-plots to visualize variability
- Geom mean honors tuning equally, insensitive to “outliers”

Probing Approach

- Many sites run periodic regression tests, e.g., nightly
  - Helps to identify performance regressions with updates
- Instead, we run a non-invasive benchmark (a probe) with a high frequency
  - Mimic the user-visible client behavior
  - Measuring latency for metadata and data operations
- Generate and analyze generated statistics
- Derive a slowdown factor (file system load)

Probing: Performance Measurement

Preparation

- Data: Generate a large file (e.g., $>$ 4x main memory of the client)
- Metadata: Pre-create a large pool of small files (e.g., 100k+ files)

Benchmarks

- Repeat the execution of the two patterns every second
- DD: Read/Write a random 1 MB block
- MD-Workbench: stat, read, delete, write a single file per iteration
  - Allows regression testing, i.e., retain the number of files
  - *J. Kunkel, G. Markomanolis. Understanding Metadata Latency with MDWorkbench.*

Executed as Bash script or an integrated tool:
`https://github.com/joobog/io-probing`

Test Systems

- JASMIN, the data analysis facility of the UK
  - Precreation: 200k files, 200 GB data file
  - 60 days of data
  - Script runs exclusively on a node
- Archer, the UK national supercomputer service
  - Precreation: 200k files, 200 GB data file
  - 30 days of data
  - Script runs on a shared interactive node
- Mistral, the HPC system at the German Climate Computing Center
  - Precreation: 100k files, 1.3 TB data file
  - 18 days of data
  - Tool runs on a shared interactive node

Understanding the Timeseries

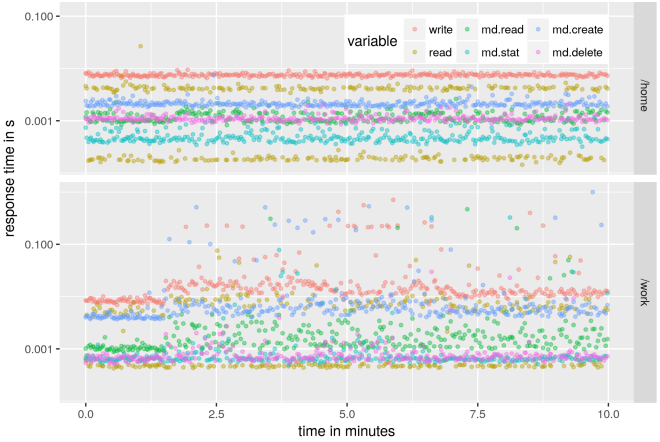


Figure: Jasmin every data point for 10 minutes of one node

- Every probe (1s) for 10 min
- For two file systems
- Home is stable
- Work shows irregularities

IO-500 Response Time on Archer

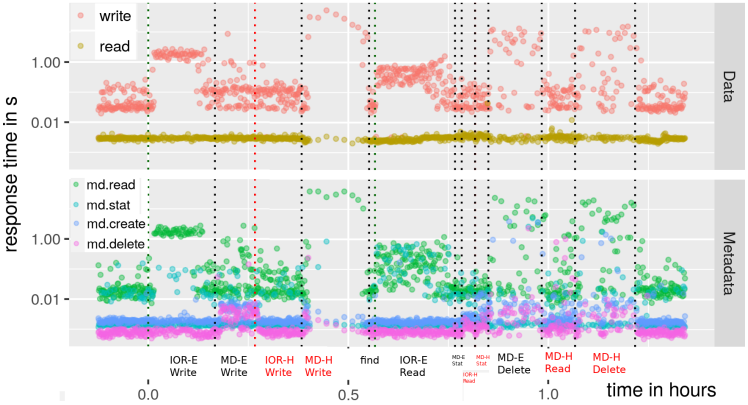


Figure: Response time (all measurements)

- Run on 100 nodes
  score 8.45
- The IO-500 various phases
  Data and metadata heavy
- First, all measurements

Validating Slowdown on All Measurements

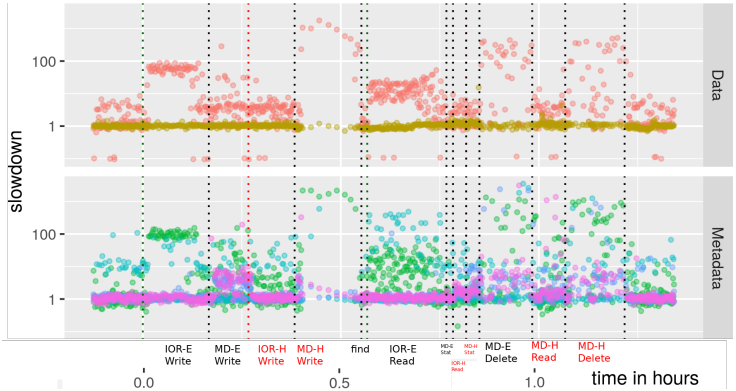


Figure: Slowdown (all measurements)

- Computed median slowdown
  Expected: median of 30 days
- Influence of phases is visible
- MDHard 1000x slowdown
  Influences data latency!
  10s of seconds latency
- IOREasy 100x slowdown
- IORHard not too much
- Data read is stable

Validating Slowdown: Reduced Data

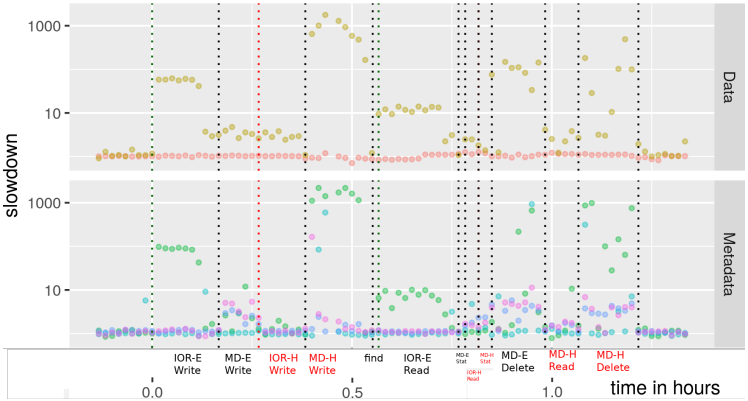


Figure: Slowdown (60s mean statistics)

- Data reduction: 60s mean
- More robust, clearer to see

Timelines of 4h Statistics

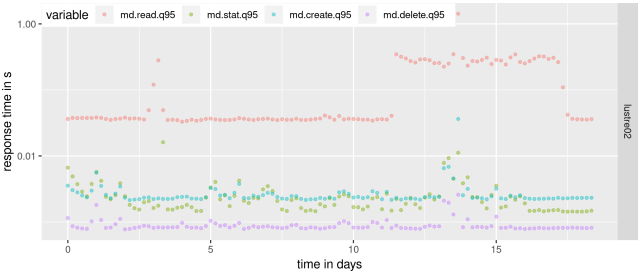


Figure: Mistral metadata timeline

- Use Q95, 5% ops are slower
- Change in behavior at day 12
  Reason: unknown

Slowdown for 4h Statistics

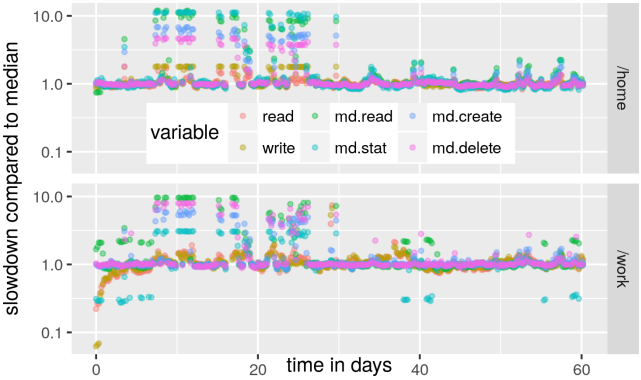
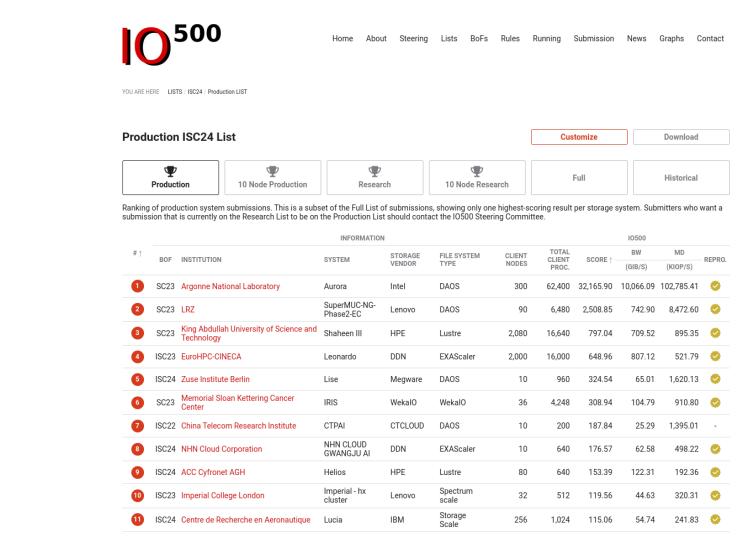


Figure: JASMIN, computed on 4 hour intervals

- Slowdown: Using the median
- Typically value is 1
- Sometimes a system is 10x slower
  - Due to user interactions
  - Concurrent application execution
- Values below 1, unusual (caching)
- Good to see long-term issues

IO500 production list



IO500

Home About Steering Lists BoFs Rules Running Submission News Graphs Contact

YOU ARE HERE LISTS / ISC24 / Production LIST

Production ISC24 List

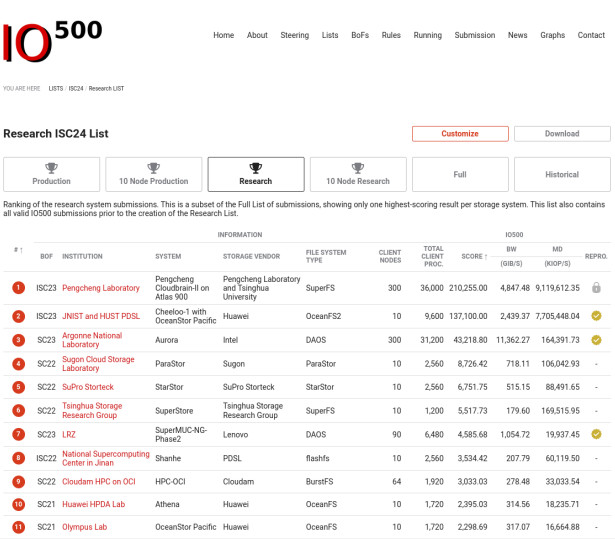
Customize Download

Production | 10 Node Production | Research | 10 Node Research | Full | Historical

Ranking of production system submissions. This is a subset of the Full List of submissions, showing only one highest-scoring result per storage system. Submitters who want a submission that is currently on the Research List to be on the Production List should contact the IO500 Steering Committee.

# ↑			INFORMATION				IO500				
	BOF	INSTITUTION	SYSTEM	STORAGE VENDOR	FILE SYSTEM TYPE	CLIENT NODES	TOTAL CLIENT PROC.	SCORE ↑	BW (GIB/S)	MD (KIOP/S)	REPRO.
1	SC23	Argonne National Laboratory	Aurora	Intel	DAOS	300	62,400	32,165.90	10,066.09	102,785.41	✓
2	SC23	LRZ	SuperMUC-NG-Phase2-EC	Lenovo	DAOS	90	6,480	2,508.85	742.90	8,472.60	✓
3	SC23	King Abdullah University of Science and Technology	Shaheen III	HPE	Lustre	2,080	16,640	797.04	709.52	895.35	✓
4	ISC23	EuroHPC-CINECA	Leonardo	DDN	EXAScaler	2,000	16,000	648.96	807.12	521.79	✓
5	ISC24	Zuse Institute Berlin	Lise	Megware	DAOS	10	960	324.54	65.01	1,620.13	✓
6	SC23	Memorial Sloan Kettering Cancer Center	IRIS	WekaIO	WekaIO	36	4,248	308.94	104.79	910.80	✓
7	ISC22	China Telecom Research Institute	CTPAI	CTCLOUD	DAOS	10	200	187.84	25.29	1,395.01	-
8	ISC24	NHN Cloud Corporation	NHN CLOUD GWANGJU AI	DDN	EXAScaler	10	640	176.57	62.58	498.22	✓
9	ISC24	ACC Cyfronet AGH	Helios	HPE	Lustre	80	640	153.39	122.31	192.36	✓
10	ISC23	Imperial College London	Imperial - hx cluster	Lenovo	Spectrum scale	32	512	119.56	44.63	320.31	✓
11	ISC24	Centre de Recherche en Aeronautique	Lucia	IBM	Storage Scale	256	1,024	115.06	54.74	241.83	✓

IO500 research list



IO500

Home About Steering Lists BoFs Rules Running Submission News Graphs Contact

YOU ARE HERE LISTS / ISC24 / Research LIST

Research ISC24 List

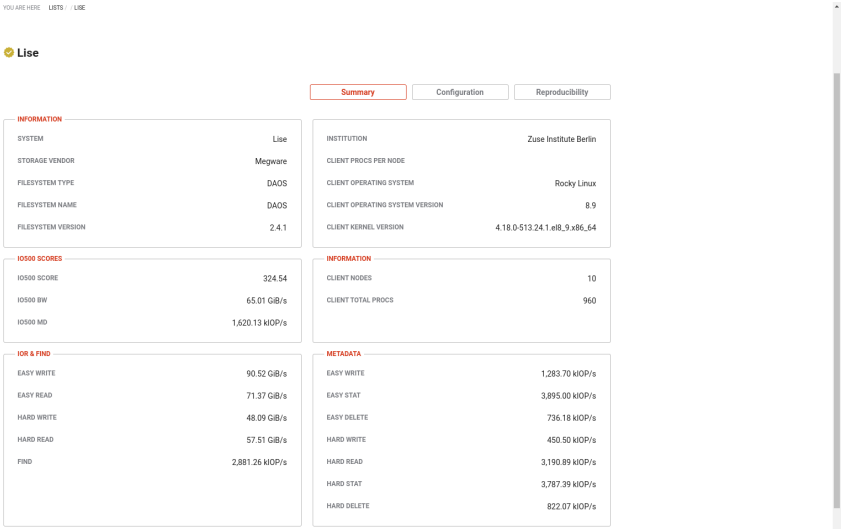
Customize Download

Production | 10 Node Production | Research | 10 Node Research | Full | Historical

Ranking of the research system submissions. This is a subset of the Full List of submissions, showing only one highest-scoring result per storage system. This list also contains all valid IO500 submissions prior to the creation of the Research List.

# ↑	BOF	INSTITUTION	SYSTEM	STORAGE VENDOR	FILE SYSTEM TYPE	CLIENT NODES	TOTAL CLIENT PROC.	IO500 SCORE ↑	BW (GIB/S)	MD (KIOP/S)	REPRO.
1	ISC23	Pengcheng Laboratory	Pengcheng Cloudbrain-II on Atlas 900	Pengcheng Laboratory and Tsinghua University	SuperFS	300	36,000	210,255.00	4,847.48	9,119,612.35	
2	ISC23	JNIST and HUST PDSL	Cheeloo-1 with OceanStor Pacific	Huawei	OceanFS2	10	9,600	137,100.00	2,439.37	7,705,448.04	
3	SC23	Argonne National Laboratory	Aurora	Intel	DAOS	300	31,200	43,218.80	11,362.27	164,391.73	
4	SC22	Sugon Cloud Storage Laboratory	ParaStor	Sugon	ParaStor	10	2,560	8,726.42	718.11	106,042.93	-
5	SC22	SuPro Storteck	StarStor	SuPro Storteck	StarStor	10	2,560	6,751.75	515.15	88,491.65	-
6	SC22	Tsinghua Storage Research Group	SuperStore	Tsinghua Storage Research Group	SuperFS	10	1,200	5,517.73	179.60	169,515.95	-
7	SC23	LRZ	SuperMUC-NG-Phase2	Lenovo	DAOS	90	6,480	4,585.68	1,054.72	19,937.45	
8	ISC22	National Supercomputing Center in Jinan	Shanhe	PDSL	flashfs	10	2,560	3,534.42	207.79	60,119.50	-
9	SC22	Cloudam HPC on OCI	HPC-OCI	Cloudam	BurstFS	64	1,920	3,033.03	278.48	33,033.54	-
10	SC21	Huawei HPDA Lab	Athena	Huawei	OceanFS	10	1,720	2,395.03	314.56	18,235.71	-
11	SC21	Olympus Lab	OceanStor Pacific	Huawei	OceanFS	10	1,720	2,298.69	317.07	16,664.88	-

IO500 result lise (ZIB)



YOU ARE HERE LISTS / LISE

Lise

Summary | Configuration | Reproducibility

INFORMATION

SYSTEM	Lise
STORAGE VENDOR	Megware
FILESYSTEM TYPE	DAOS
FILESYSTEM NAME	DAOS
FILESYSTEM VERSION	2.4.1

INSTITUTION	Zuse Institute Berlin
CLIENT PROCS PER NODE	
CLIENT OPERATING SYSTEM	Rocky Linux
CLIENT OPERATING SYSTEM VERSION	8.9
CLIENT KERNEL VERSION	4.18.0-513.24.1.el8_9.x86_64

IO500 SCORES

IO500 SCORE	324.54
IO500 BW	65.01 GiB/s
IO500 MD	1,620.13 kIOP/s

INFORMATION

CLIENT NODES	10
CLIENT TOTAL PROCS	960

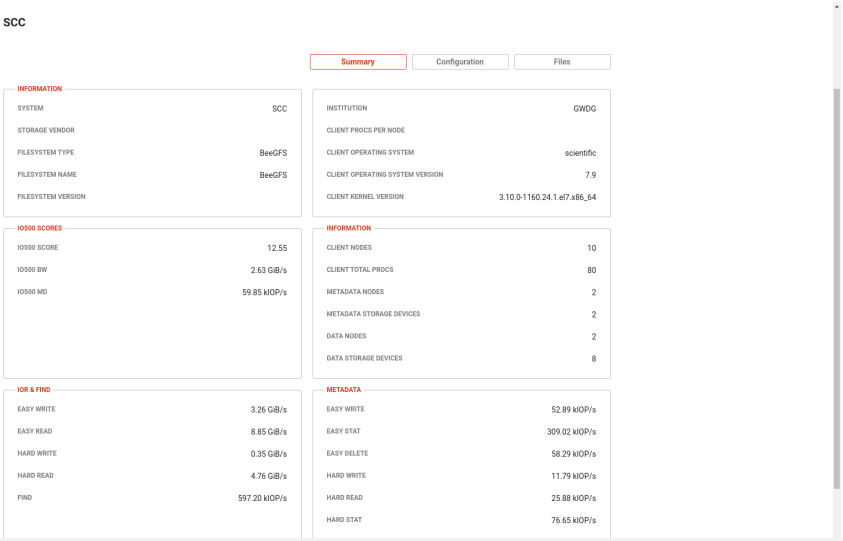
IOR & FIND

EASY WRITE	90.52 GiB/s
EASY READ	71.37 GiB/s
HARD WRITE	48.09 GiB/s
HARD READ	57.51 GiB/s
FIND	2,881.26 kIOP/s

METADATA

EASY WRITE	1,283.70 kIOP/s
EASY STAT	3,895.00 kIOP/s
EASY DELETE	736.18 kIOP/s
HARD WRITE	450.50 kIOP/s
HARD READ	3,190.89 kIOP/s
HARD STAT	3,787.39 kIOP/s
HARD DELETE	822.07 kIOP/s

IO500 result SCC (GWDG)



SCC

Summary | Configuration | Files

INFORMATION

SYSTEM	SCC
STORAGE VENDOR	
FILESYSTEM TYPE	BeeGFS
FILESYSTEM NAME	BeeGFS
FILESYSTEM VERSION	

INSTITUTION	GWDG
CLIENT PROCS PER NODE	
CLIENT OPERATING SYSTEM	scientific
CLIENT OPERATING SYSTEM VERSION	7.9
CLIENT KERNEL VERSION	3.10.0-1160.24.1.el7.x86_64

IO500 SCORES

IO500 SCORE	12.55
IO500 BW	2.63 GiB/s
IO500 MD	59.85 kIOP/s

INFORMATION

CLIENT NODES	10
CLIENT TOTAL PROCS	80
METADATA NODES	2
METADATA STORAGE DEVICES	2
DATA NODES	2
DATA STORAGE DEVICES	8

IOR & FIND

EASY WRITE	3.26 GiB/s
EASY READ	8.85 GiB/s
HARD WRITE	0.35 GiB/s
HARD READ	4.76 GiB/s
FIND	597.20 kIOP/s

METADATA

EASY WRITE	52.89 kIOP/s
EASY STAT	309.02 kIOP/s
EASY DELETE	58.29 kIOP/s
HARD WRITE	11.79 kIOP/s
HARD READ	25.88 kIOP/s
HARD STAT	76.65 kIOP/s

Summary

- Benchmarking is important for comparing systems
- Benchmarking can show problems with the storage system
- IO500 is an established standard including many different workloads
- Running IO500 can be done by anyone