IO500 based Model Exploration: From IO500 List to Clusters Information

Radita Liem[*], **Julian Kunkel**[‡], **Jay Lofstead**[†], **Christian Terboven**[*]

[*]Chair for High Performance Computing, IT Center, RWTH Aachen University
[‡]Göttingen University/GDWG
[†]Sandia National Laboratories

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737 | Sandia National Laboratories | hpc High Performance Computing | IT Center | RWTH AACHEN UNIVERSITY

Motivation

I/O performance in a multi-user environment is difficult to predict. It is hard for users to know what to expect when they are running and tuning their application for better I/O performance.

We propose to explore **IO500 benchmark** as a way to guide users to get **realistic expectations** about their application's I/O and optimization strategy based on the cluster's capability.

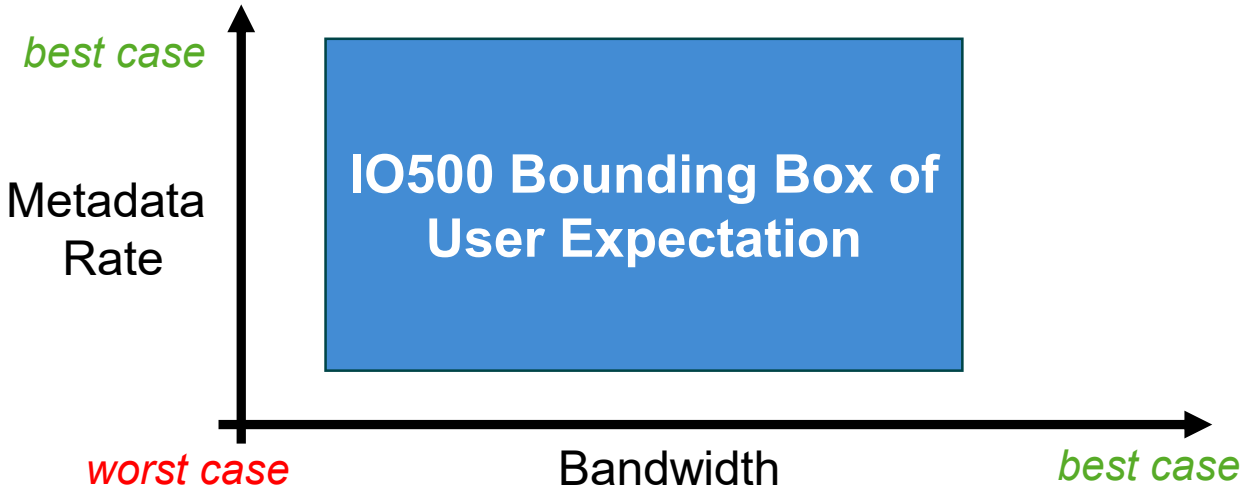
IO500 Benchmark Overview

- Developed in 2017 and quickly becoming the de facto benchmarking standard for HPC Storage[2].
- Designed to create balanced performance measurement
- The benchmark has 5 scenarios[3] :
  - **IOR 'easy':** Free to tune IOR parameters. Typically file-per-process, large, aligned chunks. Creating **best possible bandwidth performance**.
  - **IOR 'hard':** Limited options to tune. Forced to use small unaligned I/O to a single shared file. Providing **worst possible bandwidth performance**.
  - **mdtest 'easy':** Free to tune mdtest parameters with zero size files in separate directory per process. Creating **best case scenario for metadata rate**
  - **mdtest 'hard':** Limited options to tune. Forced all processes to write into a single shared directory. Providing **worst case scenario for metadata rate**.
  - **Find:** Finding specific subset of files created by 4 other benchmarks.
- The numbers from the scenarios are combined using geometric mean and each scenario is running for 300s that eliminates the cache effect

[2] J. Kunkel, "Virtual Institute for I/O," *ISC'19*: The IO-500 and the Virtual Institute of I/O, 30-Oct-2020. [Online]. Available: https://www.vi4io.org/io500/bofs/isc19/start. [Accessed: 27-May-2021].
[3] M. Rásó-Barnett, "Lustre and IO-500: Experiences with the Cambridge Data Accelerator", 2019. [Online]. https://www.eofs.eu/_media/events/lad19/03_matt_raso-barnett-io500-cambridge.pdf. [Accessed: 02-Mar-2021]

IO500 Benchmark Usage

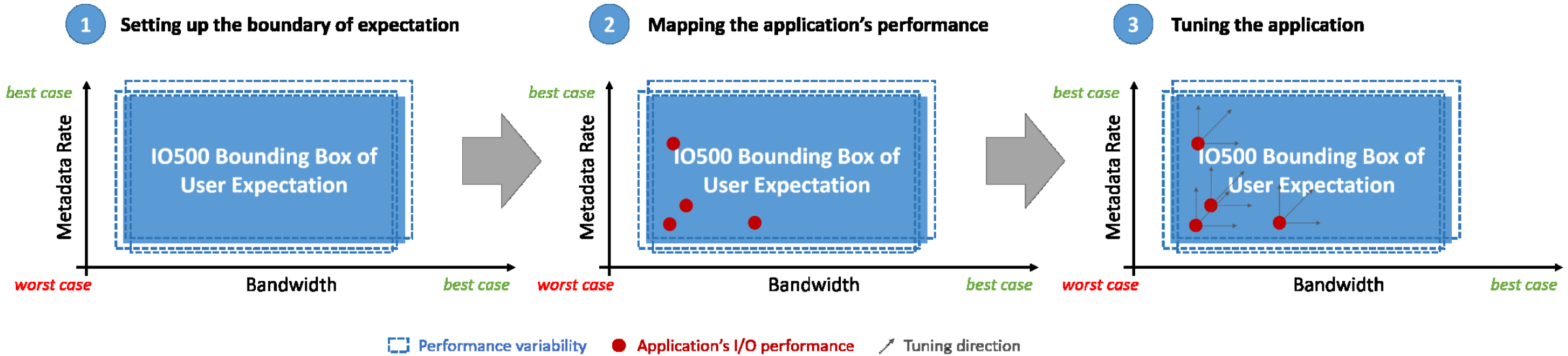
- IO500 benchmark's mdtest and IOR scenario can be used to form a bounding box of user expectations [4] as illustrated by the figure below



- **Worst case scenario** is coming from IOR and mdtest **'hard'** scenario
  Best case scenario is coming from IOR and mdtest '**easy**' scenario
- **'Find'** is not used in this bounding box model since it is not as controlled as IOR and mdtest and will skew the IO500 numbers

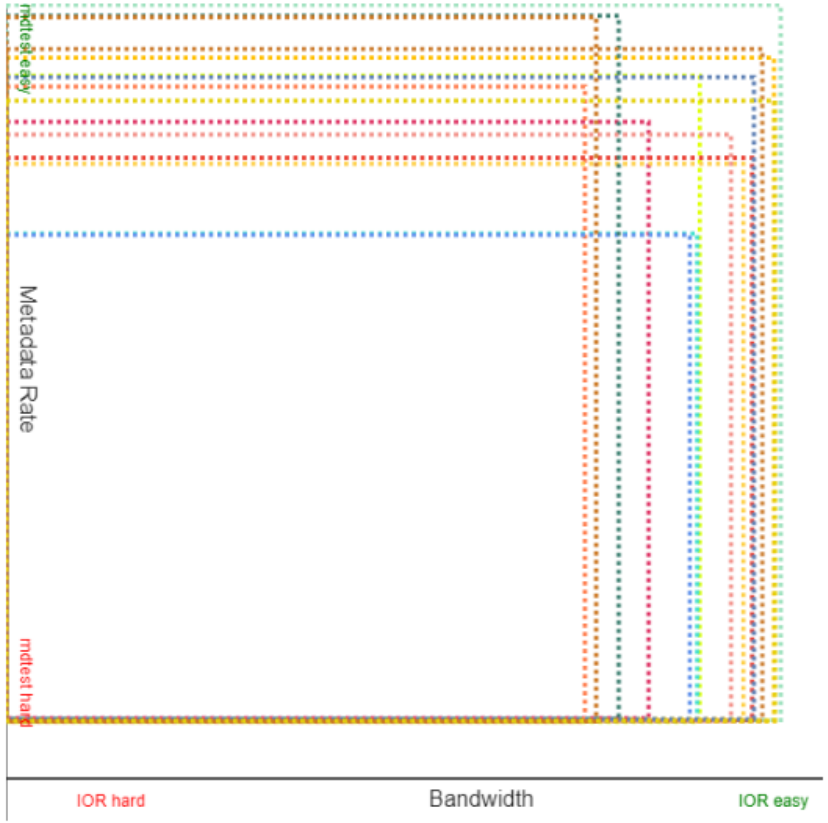
[4] A. Dilger, "IO500 | A storage Benchmark for HPC", 2019. [Online]. Available:https://wiki.lustre.org/images/9/92/LUG2019-IO500_Storage_Benchmark_for_HPC-Dilger.pdf. [Accessed: 02-Mar-2021]

Bounding Box of User Expectation (BBoUE) Workflow



Results: Forming Bounding Box of User Expectation

- Bounding box of **POSIX** API, each square represents individual run from the same IO configuration



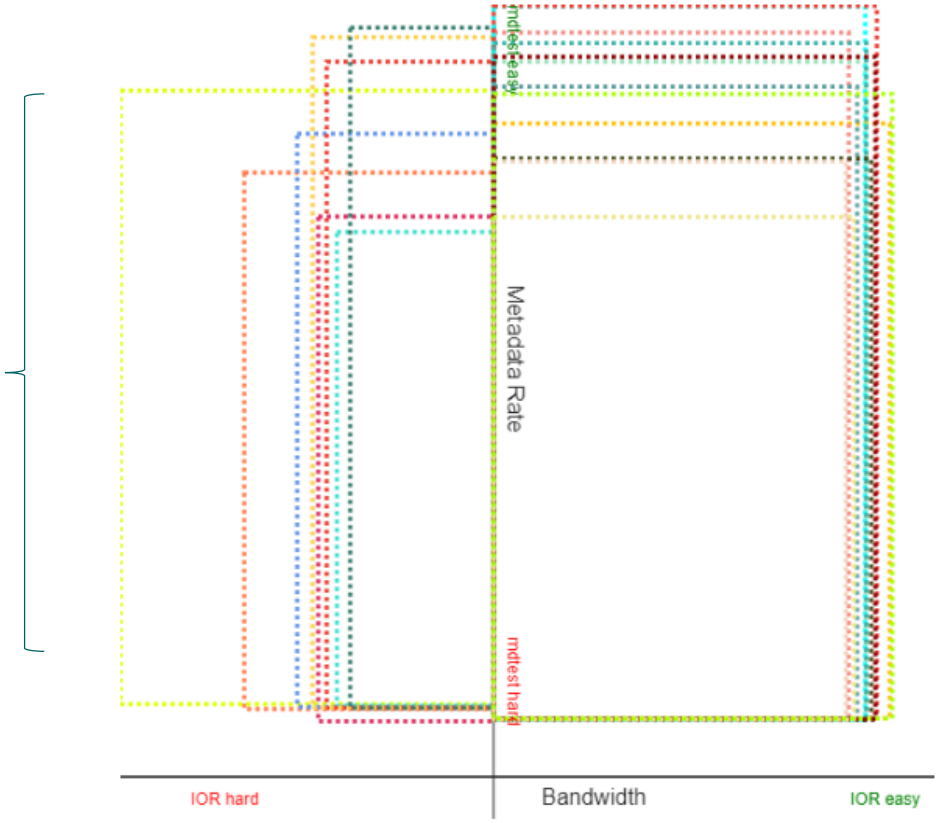
	IOR hard (GiB/s)	IOR easy (GiB/s)	mdtest hard (KIOPS)	mdtest easy (KIOPS)
	0.85	1.88	10.97	145.17
	1.03	1.89	11.26	123.26
	1.10	1.87	10.59	129.86
	0.87	1.90	10.76	135.27
	0.97	1.90	10.64	131.93
	0.94	1.86	11.06	102.35
	0.95	1.86	10.84	102.06
	0.90	1.89	10.98	116.54
	0.90	1.89	11.16	115.40
	1.08	1.90	11.14	143.09
	0.91	1.88	10.80	120.86
	0.90	1.90	11.05	131.65
	0.87	1.88	10.79	136.91

This project is currently displayed in:https://bit.ly/3BhhAFZ

Results: Anomalous Bounding Box

- Anomalous result in **MPI-IO** API: IOR ‘Easy’ score gets lower number than IOR ‘hard’
- Broken node is most likely the reason behind these anomalous result

Bounding box skewed
to the direction of IOR ‘hard’



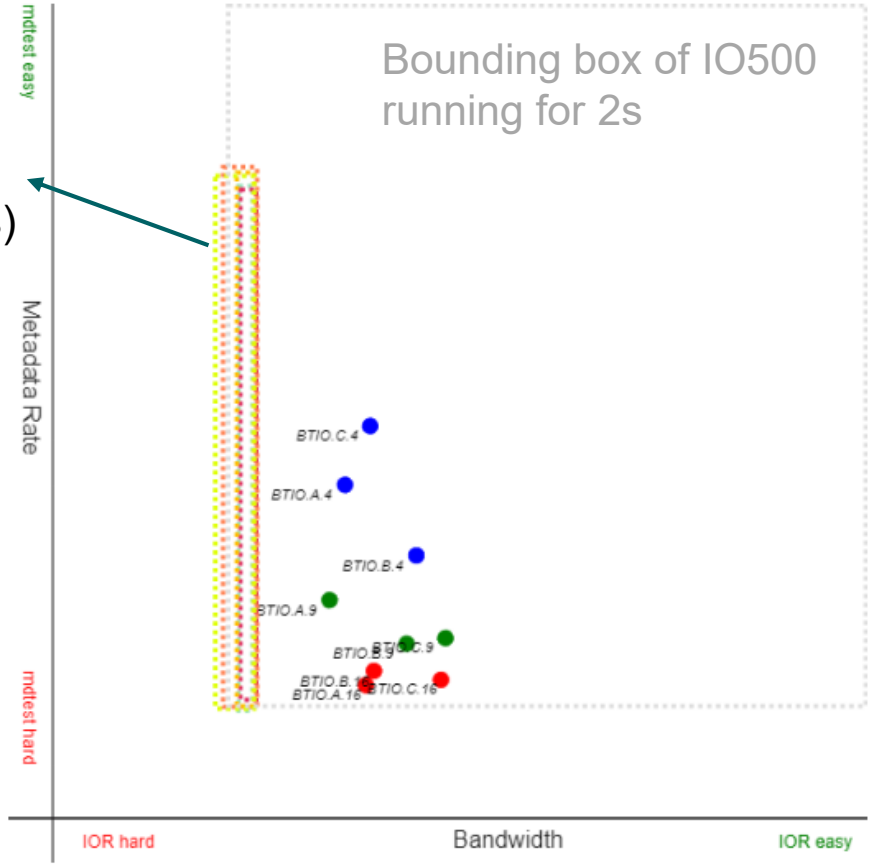
	IOR hard (GiB/s)	IOR easy (GiB/s)	mdtest hard (KIOPS)	mdtest easy (KIOPS)
	0.90	1.89	10.74	135.79
	0.94	0.47	10.50	106.45
	1.14	0.47	12.86	114.73
	0.83	1.89	11.12	124.06
	1.46	0.47	13.78	130.29
	0.88	0.47	13.50	103.47
	0.99	0.47	13.24	122.10
	0.91	0.47	13.04	135.73
	0.95	0.46	13.10	140.41
	0.85	0.47	13.02	142.20
	0.96	1.90	11.00	141.28
	0.86	1.85	10.81	146.24
	0.91	1.87	11.03	131.02

This project is currently displayed in:https://bit.ly/3BhhAFZ

Results: Mapping I/O Performance with Darshan

- Exploration with BTIO shows the application's performance falls within the box for **MPI-IO API *with cache effect***

Bounding boxes of IO500 running with default setup (300s)

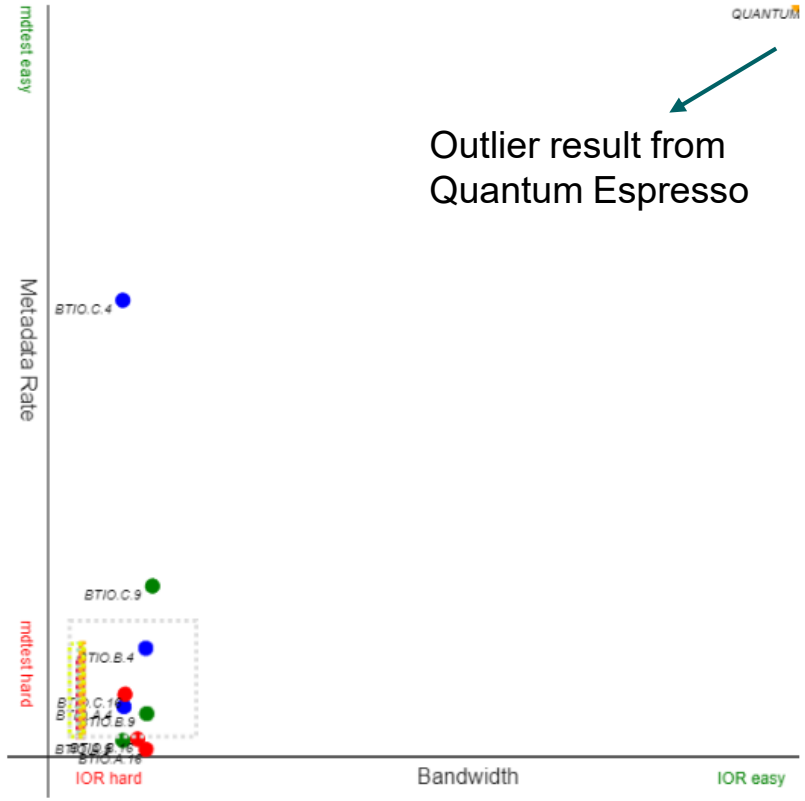


	Benchmark	Bandwidth (GiB/s)	Metadata Rate (KIOPS)
	BTIO.A.4	0.68	21.67
	BTIO.A.9	0.64	14.18
	BTIO.A.16	0.73	8.66
	BTIO.B.4	0.85	17.08
	BTIO.B.9	0.82	11.35
	BTIO.B.16	0.75	9.58
	BTIO.C.4	0.74	25.50
	BTIO.C.9	0.91	11.71
	BTIO.C.16	0.90	9.00

This project is currently displayed in:https://bit.ly/3BhhAFZ

Results: Mapping I/O Performance with Darshan - Outlier Result

- However in the **POSIX API**, metadata rate calculation falls outside the cache box and outlier result for Quantum Espresso



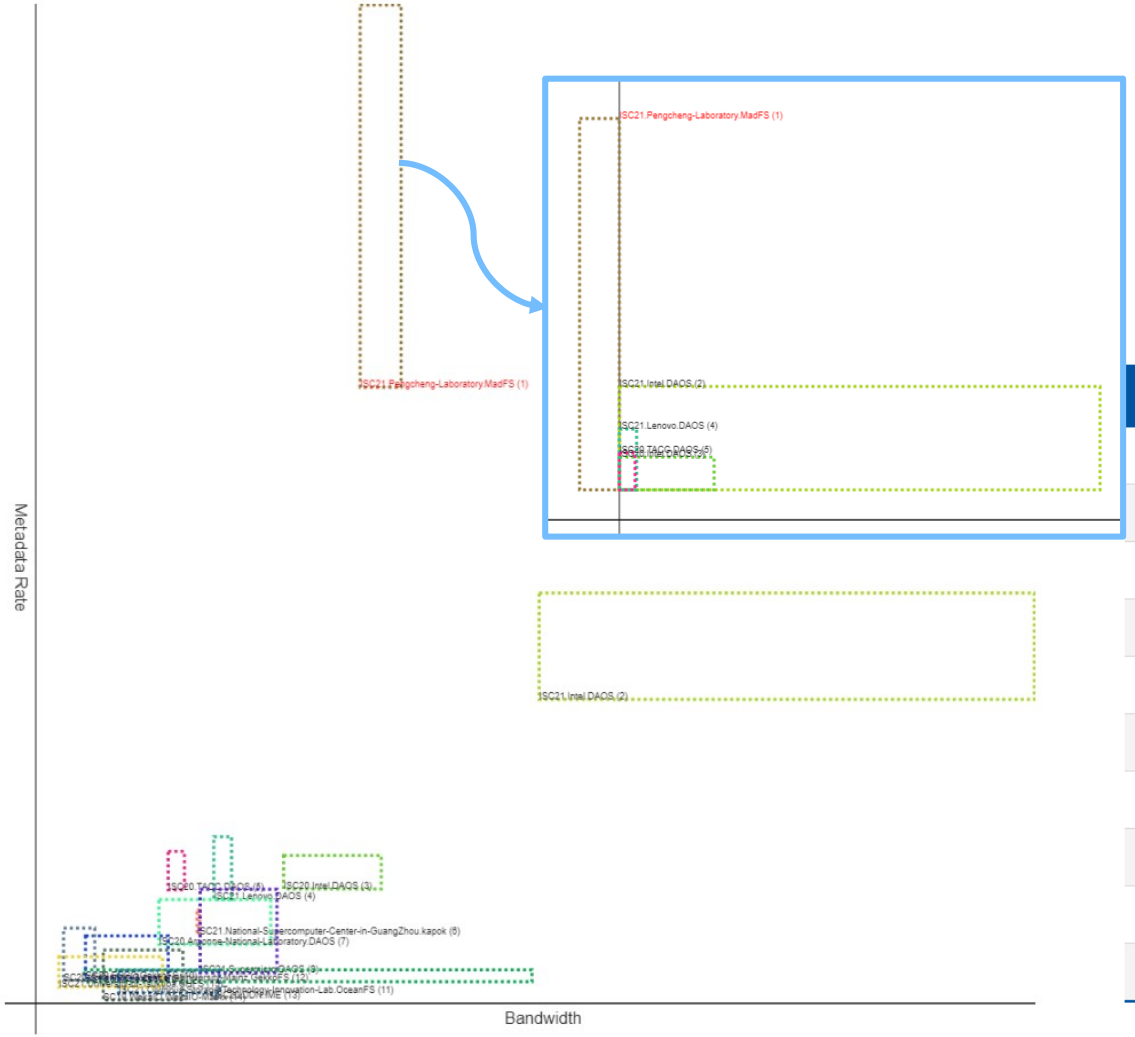
Outlier result from Quantum Espresso

	Benchmark	Bandwidth (GiB/s)	Metadata Rate (KIOPS)
	BTIO.A.4	0.99	18.42
	BTIO.A.9	0.97	5.99
	BTIO.A.16	1.27	2.69
	BTIO.B.4	1.27	39.90
	BTIO.B.9	1.28	15.84
	BTIO.B.16	1.16	6.51
	BTIO.C.4	0.97	167.96
	BTIO.C.9	1.36	62.81
	BTIO.C.16	1.00	22.99
	QUANTUM ESPRESSO.24	9.77	276.59

Metadata in BT-IO are scattered outside the bounding box

This project is currently displayed in:https://bit.ly/3BhhAFZ

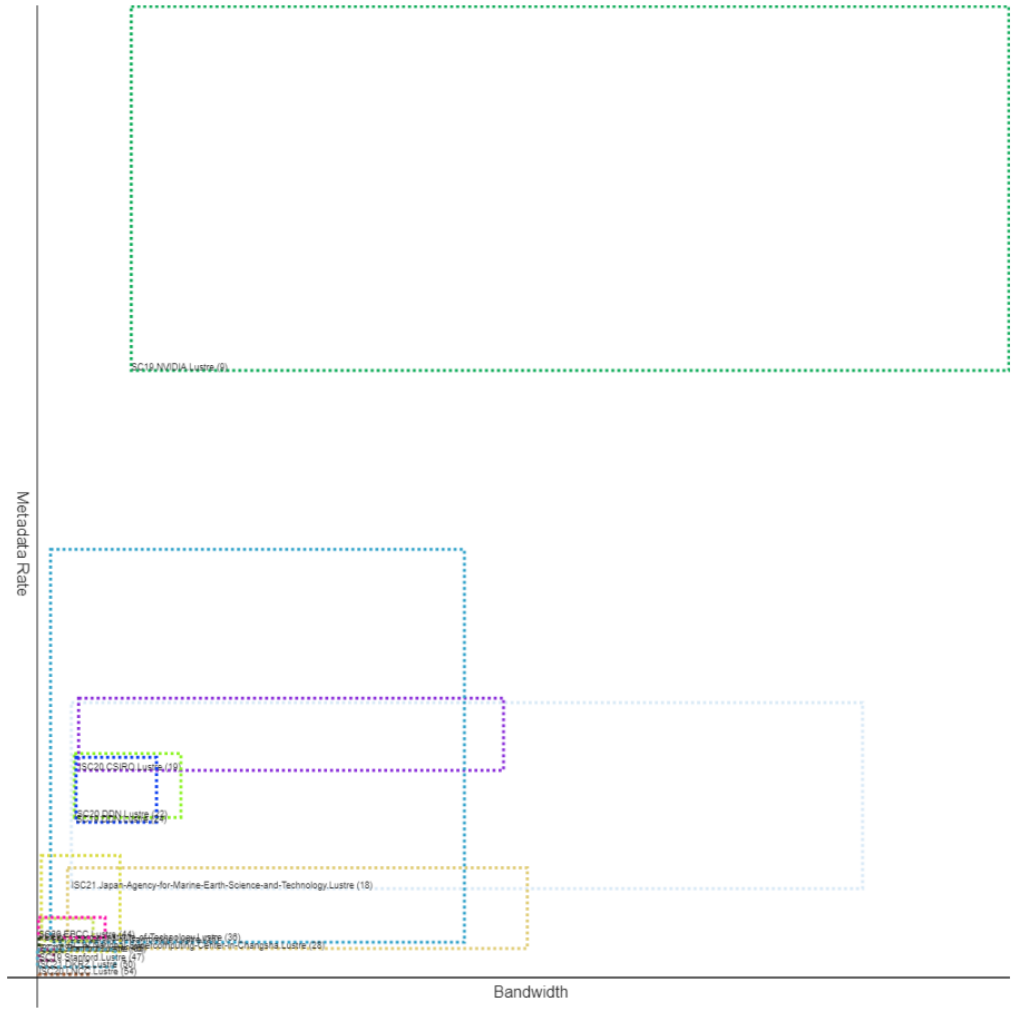
Using BBoUE on the IO500 List Result



- Interesting form of bounding box of user expectation from the top 2 of the list
- IOR hard performs better than the IOR easy in MadFS. Potential improvement?

Tag	IOR hard (GiB/s)	IOR easy (GiB/s)	mdtest hard (KIOPS)	mdtest easy (KIOPS)
ISC21.Pengcheng-Laboratory.MadFS	205.33	182.85	16380.06	26569.90
ISC21.Intel.DAOS	282.97	561.94	8081.06	10934.75
ISC20.Intel.DAOS	139.37	194.80	3038.19	3959.73
ISC21.Lenovo.DAOS	100.11	110.71	2767.18	4457.22
ISC20.TACC.DAOS	74.49	84.13	3023.32	4064.34
ISC21.National-Supercomputer-Center-in-GuangZhou.kapok	90.63	92.65	1831.38	2503.70
ISC20.Argonne-National-Laboratory.DAOS	69.21	132.61	1568.07	2786.25
ISC21.Supermicro.DAOS	92.44	136.12	815.94	3061.71
SC19.NVIDIA.Lustre	27.06	279.59	574.07	919.45
SC20.EPCC.GekkoFS	27.95	75.02	669.01	1824.42

Using BBoUE on the IO500 List Result - Lustre

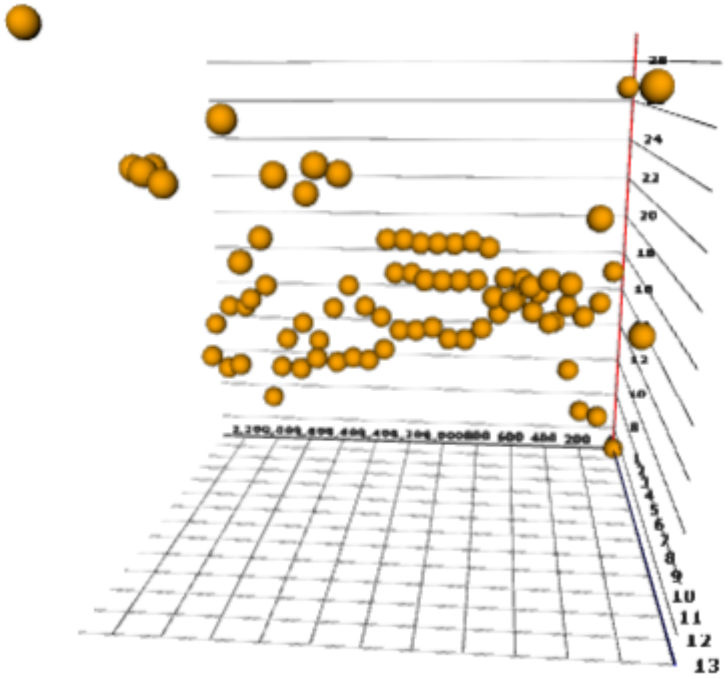


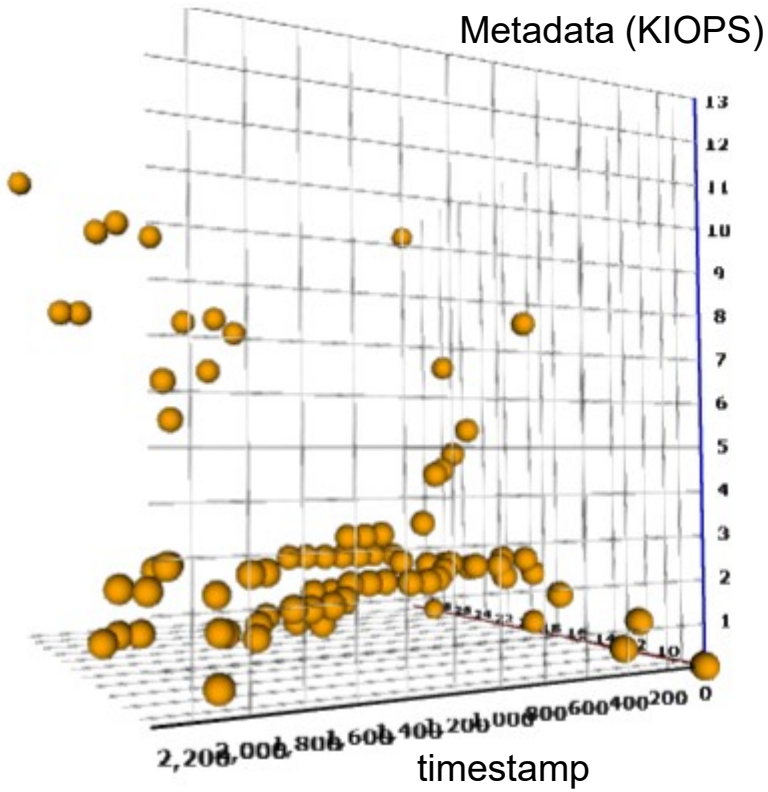
- Removing outliers from equation. Let's try testing with mainstream parallel FS: Lustre
- All same filesystems, different boxes
- What should be an ideal bounding box looks like?

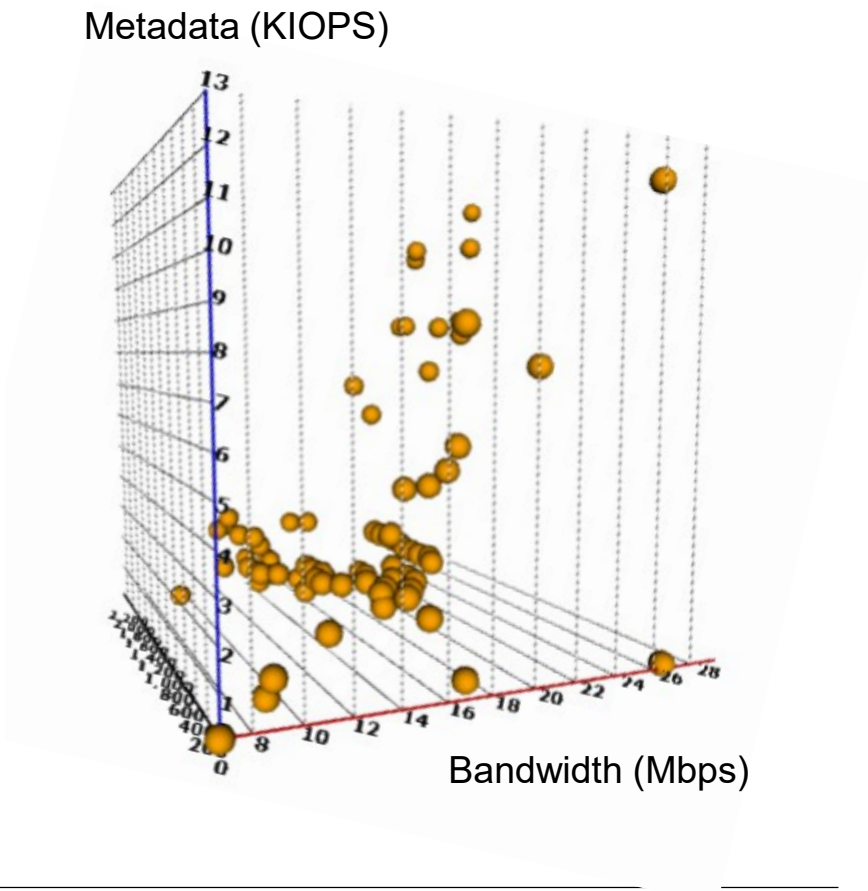
Tag	IOR hard (GiB/s)	IOR easy (GiB/s)	mdtest hard (KIOPS)	mdtest easy (KIOPS)
SC19.NVIDIA.Lustre	27.06	279.59	574.07	919.45
ISC21.Japan-Agency-for-Marine-Earth-Science-and-Technology.Lustre	9.78	237.47	84.08	260.98
ISC20.CSIRO.Lustre	11.85	134.32	196.05	265.29
ISC20.DDN.Lustre	10.63	41.54	151.41	212.77
SC19.DDN.Lustre	11.18	34.55	146.83	209.32
SC19.University-of-Cambridge.Lustre	3.83	123.05	33.34	405.92
SC19.National-Supercomputing-Center-in-Changsha.Lustre	8.71	141.16	27.41	104.74
SC19.Stanford.Lustre	1.13	24.10	24.56	116.32
ISC21.Georgia-Institute-of-Technology.Lustre	1.25	16.34	34.84	56.93
SC20.EPCC.Lustre	0.35	19.85	37.83	57.88
SC19.Stanford.Lustre	0.47	7.48	24.53	38.96
SC19.Stanford.Lustre	0.55	4.94	16.45	27.58
ISC21.DKRZ.Lustre	0.06	22.70	9.85	27.45
ISC20.LNCC.Lustre	0.35	15.68	3.25	3.70

From Bounding Box to Bounding Cube

- Integrating time and phase aspect to the performance monitoring
- Initial prototype: Combining Metadata and Bandwidth Information from the system monitoring
- Sample using PIKA output from Dresden







GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737 | Sandia National Laboratories | i12 High Performance Computing | IT Center | RWTH AACHEN UNIVERSITY

Thank you!

If you have any question: **Radita Liem** (liem@itc.rwth-aachen.de)