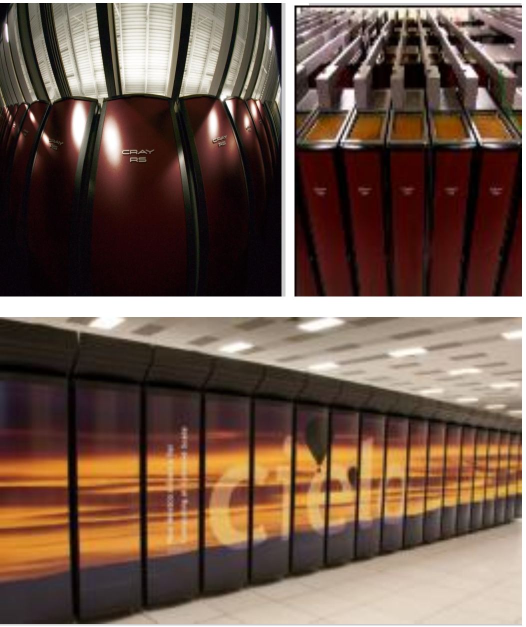


Sandia National Laboratories

Exceptional service in the national interest

Creating a Meaningless Single Number to Appease Human Irrationality

Jay Lofstead, Julian Kunkel, John Bent, George Markomanolis

Scalable System Software
Sandia National Laboratories
Albuquerque, NM, USA
gflofst@sandia.gov

Lustre Users Group

April 26, 2018

SAND2017-11869 PE

IO500

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The downsides of civilian leadership

- Non-experts have no way to compare similar things with subtle differences
  - Top500 list idea offers a way to somewhat compare computers
- Storage central feature for a platform's ability to actually generate science, but generally budgeted around 10%.
  - Hard to explain intricacies and applications each have different tradeoffs and requirements. 10% is "good enough" since Top500 drives "biggest/fastest" claims.
- How do we elevate storage to balance platforms and share best practices so we all get the best from our systems?

So a single number is needed

- Workloads matter—a lot
  - Small files vs. large, number of writers, data (re)distribution requirements
- Storage system characteristics vary—a lot
  - Hardware and placement, software
- Do we include all possible workloads?
  - Data analytics are read intensive while ModSim are write intensive
  - Data reuse (possibility of “hot” data) differs greatly.
- How do we learn from earlier efforts?
  - Previous attempts never got to the finish line
- What do we do about incalcitrant sites?
  - Declarations that if benchmarks run on machines, user will be banned

What is most important?

- DOE sites use 90% forward progress as a reliability metric. That means we can use about 6 minutes total per hour for resilience related activities.
  - So 5 minutes seems reasonable to keep things short.
- Metadata must be measured
  - 1 file per process is still in wide use
- Bandwidth must be measured
  - Can we write the percentage of memory to storage in the required time?
- The “best” system will be “balanced”
  - Can’t neglect one for the other to be good for a variety of workloads
  - “Number” must represent both somehow.

Can we include other tasks?

- Typical IO operations performed by applications are easy to model.
  - IOR and mdtest are not perfect, but accepted. Tweaks can expand applicability.
- Admin tasks for things like purges are often neglected
  - "find" files that haven't been touched in x time and/or bigger than a particular size.
- What about those pesky, increasingly important data analytics workloads?
  - Is there even an accepted benchmark that people would accept?

What about “cheaters”?

- Competitive environments tempt people to cheat
- Given we, the organizers, don’t want to closely audit everyone, how do we manage cheating?
    - Auditing too hard and time consuming
    - Access requirements may prevent effective auditing
    - Takes too much time for a side project
    - Cheaters have insights that can be useful for others
- Let’s encourage, but manage, “cheating” to help the community!

Lists are boring—but important

- Lists of stats are boring and nearly useless for research or production purposes

- Lists are crucial for documenting what exists for historical purposes

- Lists offer little value because they get stale

- Lists can offer advice to tune systems or avoid purchasing mistakes

Basic approach

- Keep the two basics
  - IOR and mdtest
- Make a pathologically bad configuration to try to reveal worst case
- Allow users to configure as they see fit to reveal how good their system can be
- Include optional pieces that explore new components
  - ”Find” and alternatives to mdtest

The Metrics

- ior easy
  - write and read
- ior hard
  - write and read
- mdtest easy
  - create, stat, delete
- mdtest hard
  - create, stat, read, delete
- ”find”

The Score

- Bandwidth
  - geo_mean of the IOR scores
- Iops
  - geo_mean of mdtest scores and “find”
- Total
  - sq_root(bandwidth* iops)

Reporting Requirements

- Include the output from the scripts
- Include as much detail as possible for the system configuration tested
- Include all details about how “easy” tests are configured

- Layered storage (e.g., burst buffers + PFS) are tested separately

- Overall ranked list is the total number, but list can be resorted based on user preferences
- “Cheating” configurations made available so surprise numbers can be tested by others

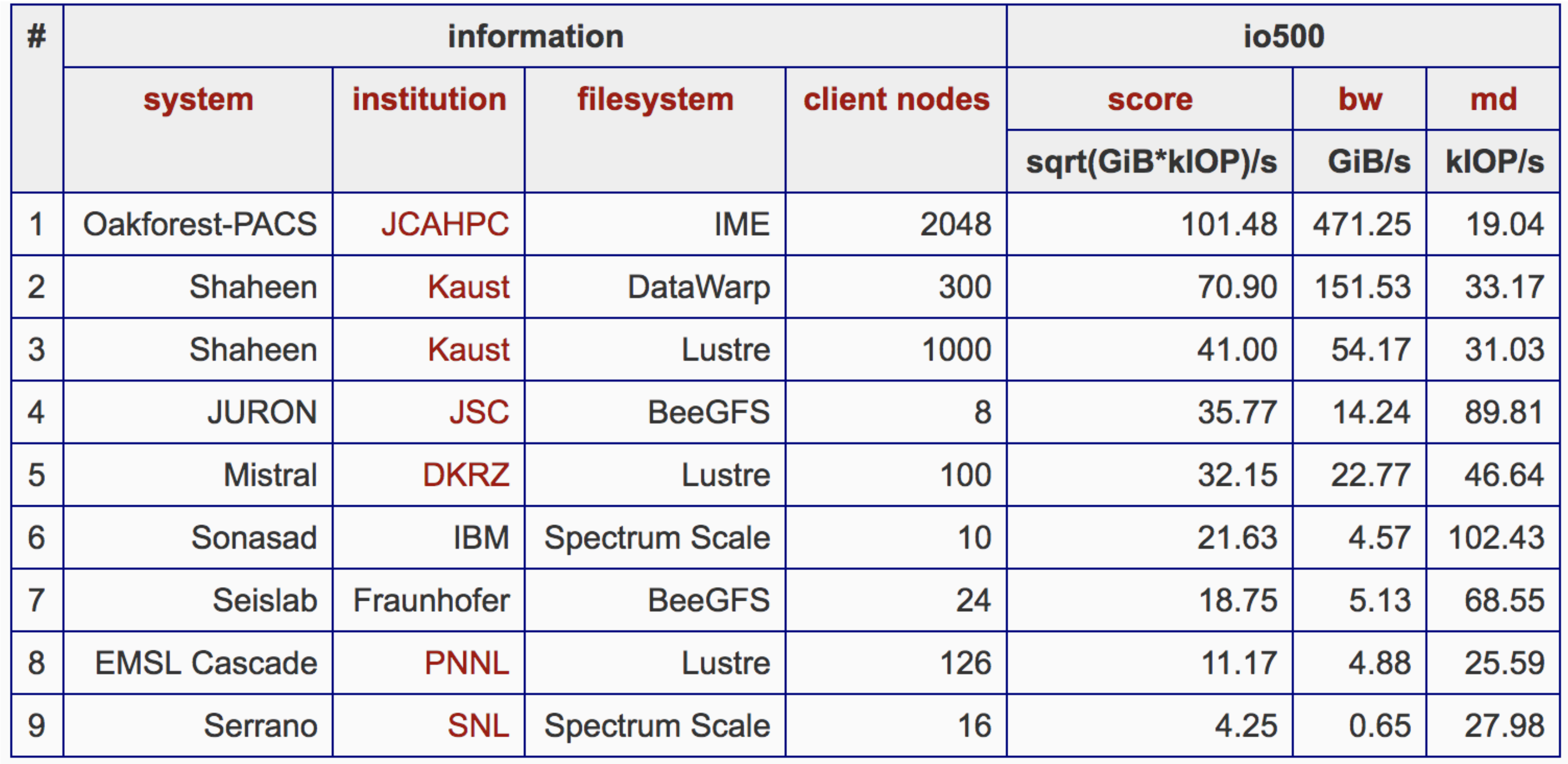
List publication

- We are sheep! :-)

- Twice a year at SC and ISC
  - First list at SC 2017, second scheduled for ISC 2018

What the first list looked like

- IME won from dominating bandwidth, but lagging IOPS



#	information				io500		
	system	institution	filesystem	client nodes	score	bw	md
					sqrt(GiB*kIOP)/s	GiB/s	kIOP/s
1	Oakforest-PACS	JCAHPC	IME	2048	101.48	471.25	19.04
2	Shaheen	Kaust	DataWarp	300	70.90	151.53	33.17
3	Shaheen	Kaust	Lustre	1000	41.00	54.17	31.03
4	JURON	JSC	BeeGFS	8	35.77	14.24	89.81
5	Mistral	DKRZ	Lustre	100	32.15	22.77	46.64
6	Sonasad	IBM	Spectrum Scale	10	21.63	4.57	102.43
7	Seislab	Fraunhofer	BeeGFS	24	18.75	5.13	68.55
8	EMSL Cascade	PNNL	Lustre	126	11.17	4.88	25.59
9	Serrano	SNL	Spectrum Scale	16	4.25	0.65	27.98

Keeping it relevant

- Top500's static test suite often criticized as not representing a lot of contemporary workloads
- Highly dynamic component set makes comparing systems year to year hard to impossible
- Improved benchmarks, accepted by the community, are desired
- New workloads, such as data analytics, strongly encouraged

- Work these in slowly after extensive vetting and community acceptance

Short History

- Virtual Institute for IO (VI4IO) created December 29, 2015
  - Julian Kunkel registered domain name
  - Open, free community for storage and IO related professionals to share knowledge and network
  - Includes catalog of storage systems around the world including benchmark results
  - Slow to gain traction with small motivation for participation
- IO-500 created June 20, 2016
  - John Bent wanted to create the competitive list
  - Natural addition to VI4IO effort adding a competition to the existing effort to motivate participation
- Quickly brought together to leverage effort

VI4IO Goals

- Document storage system design
  - Offer long-term storage system design archive, including benchmarks
- Share best practices
  - No organized approach, but desired goal
- Build community
  - No barriers to entry to encourage broad participation

- Had some difficulty gaining traction

IO 500 Goals

- Competitive list for bragging about storage systems
  - Easier to justify to management compute time to run benchmarks
- Develop Best Practices database through the benchmarks
  - Do things we know are hard and require “easy” things fully end-user configurable.
  - Must reveal how easy tests are done and submit code for any custom tools (e.g., for find)

- Natural match with VI4IO

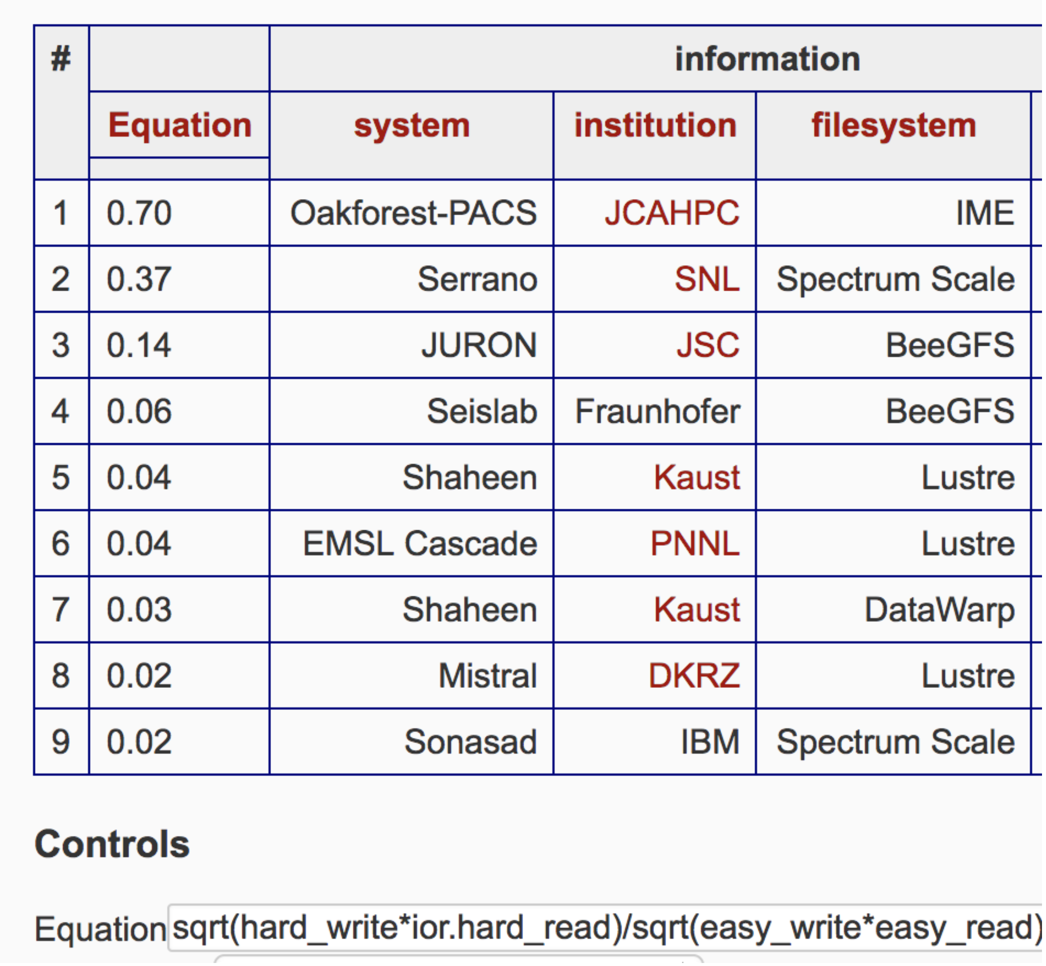
VI4IO and IO 500 Mission

Mission:

1. Provide a competitive list to justify compute time
2. Gather best practices for different storage system designs
3. Document various storage systems
4. Friendly cooperation and competition

Use accepted benchmarks using generally accepted configurations (for the hard setup)

Least degradation from IOR easy to hard

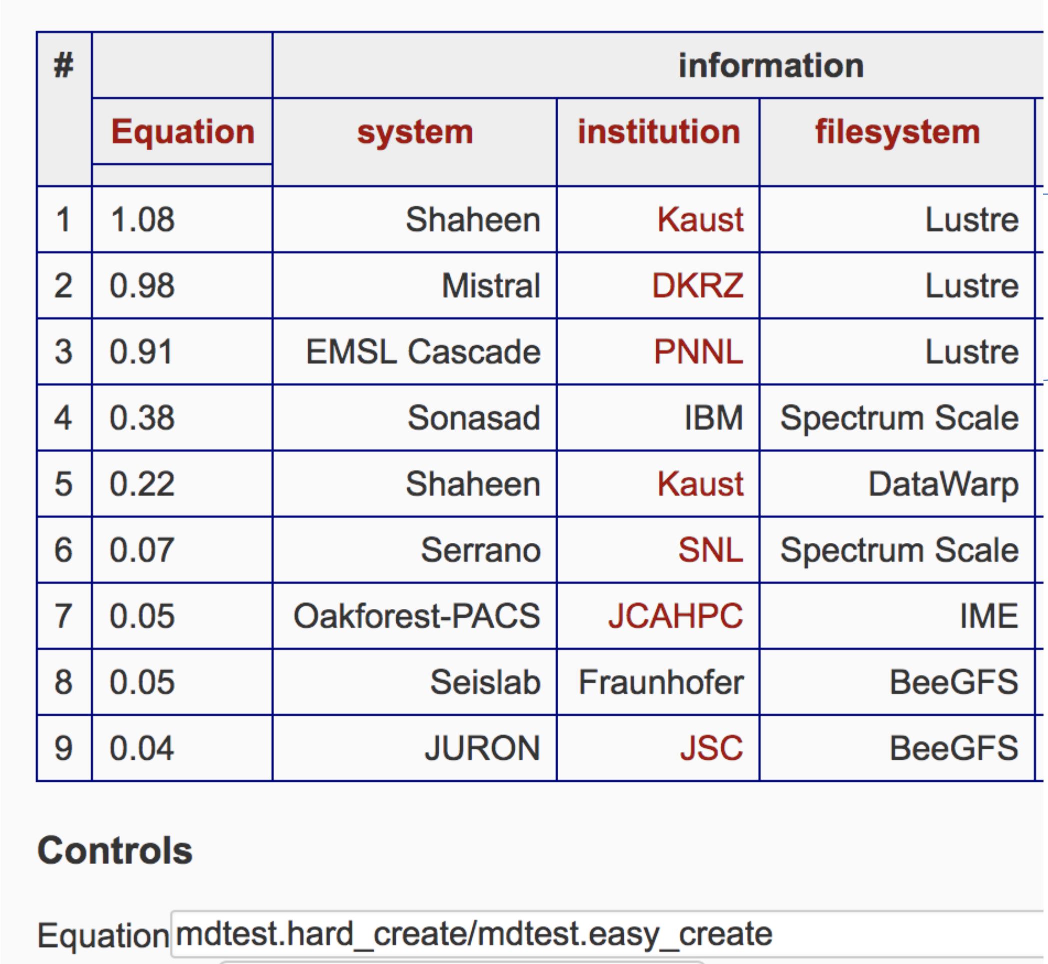


#		information		
	Equation	system	institution	filesystem
1	0.70	Oakforest-PACS	JCAHPC	IME
2	0.37	Serrano	SNL	Spectrum Scale
3	0.14	JURON	JSC	BeeGFS
4	0.06	Seislab	Fraunhofer	BeeGFS
5	0.04	Shaheen	Kaust	Lustre
6	0.04	EMSL Cascade	PNNL	Lustre
7	0.03	Shaheen	Kaust	DataWarp
8	0.02	Mistral	DKRZ	Lustre
9	0.02	Sonasad	IBM	Spectrum Scale

Controls

Equation sqrt(hard_write*ior.hard_read)/sqrt(easy_write*easy_read)

Degradation for creates in shared directory



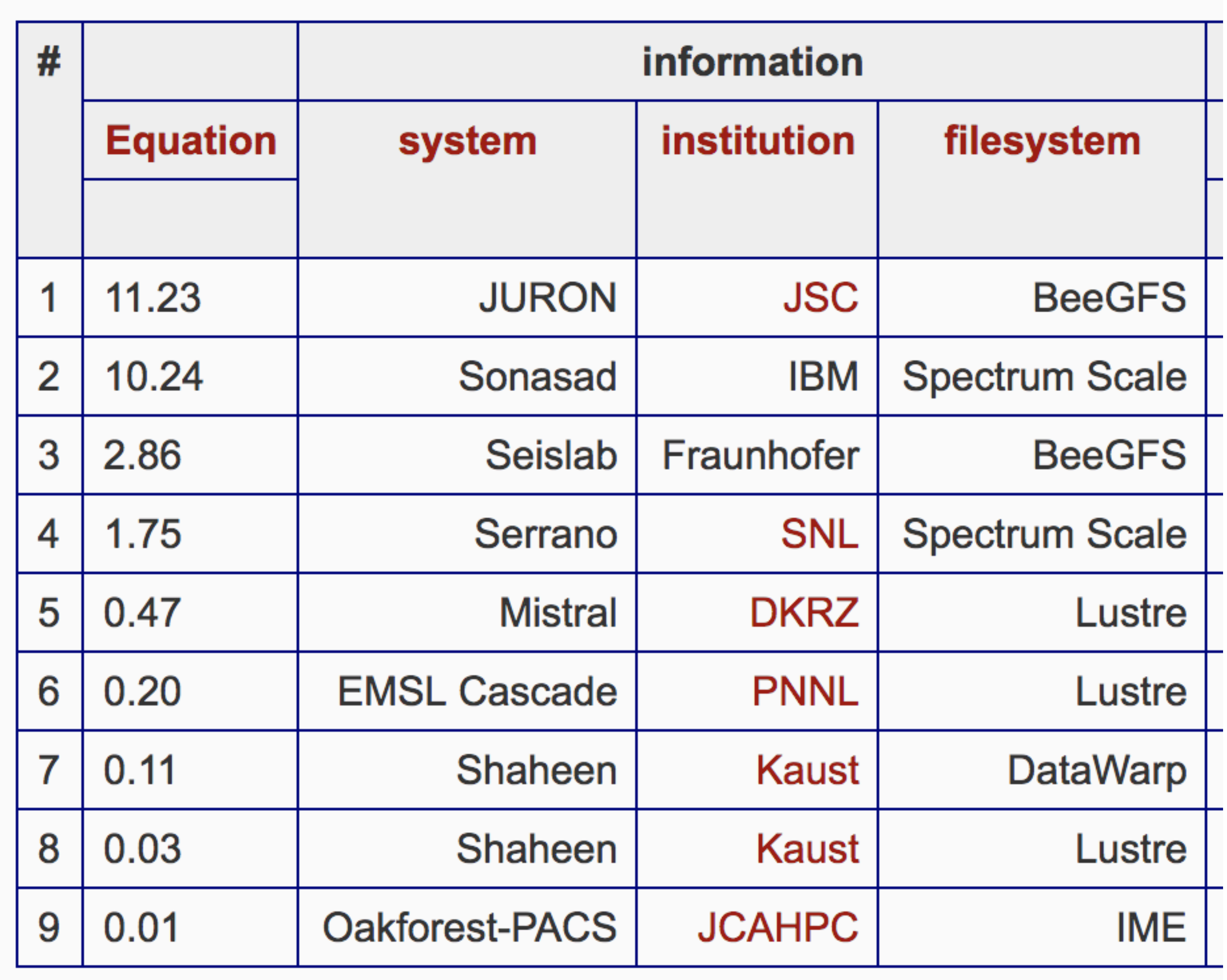
#		information		
	Equation	**system**	**institution**	**filesystem**
1	1.08	Shaheen	Kaust	Lustre
2	0.98	Mistral	DKRZ	Lustre
3	0.91	EMSL Cascade	PNNL	Lustre
4	0.38	Sonasad	IBM	Spectrum Scale
5	0.22	Shaheen	Kaust	DataWarp
6	0.07	Serrano	SNL	Spectrum Scale
7	0.05	Oakforest-PACS	JCAHPC	IME
8	0.05	Seislab	Fraunhofer	BeeGFS
9	0.04	JURON	JSC	BeeGFS

Controls

Equation mdtest.hard_create/mdtest.easy_create

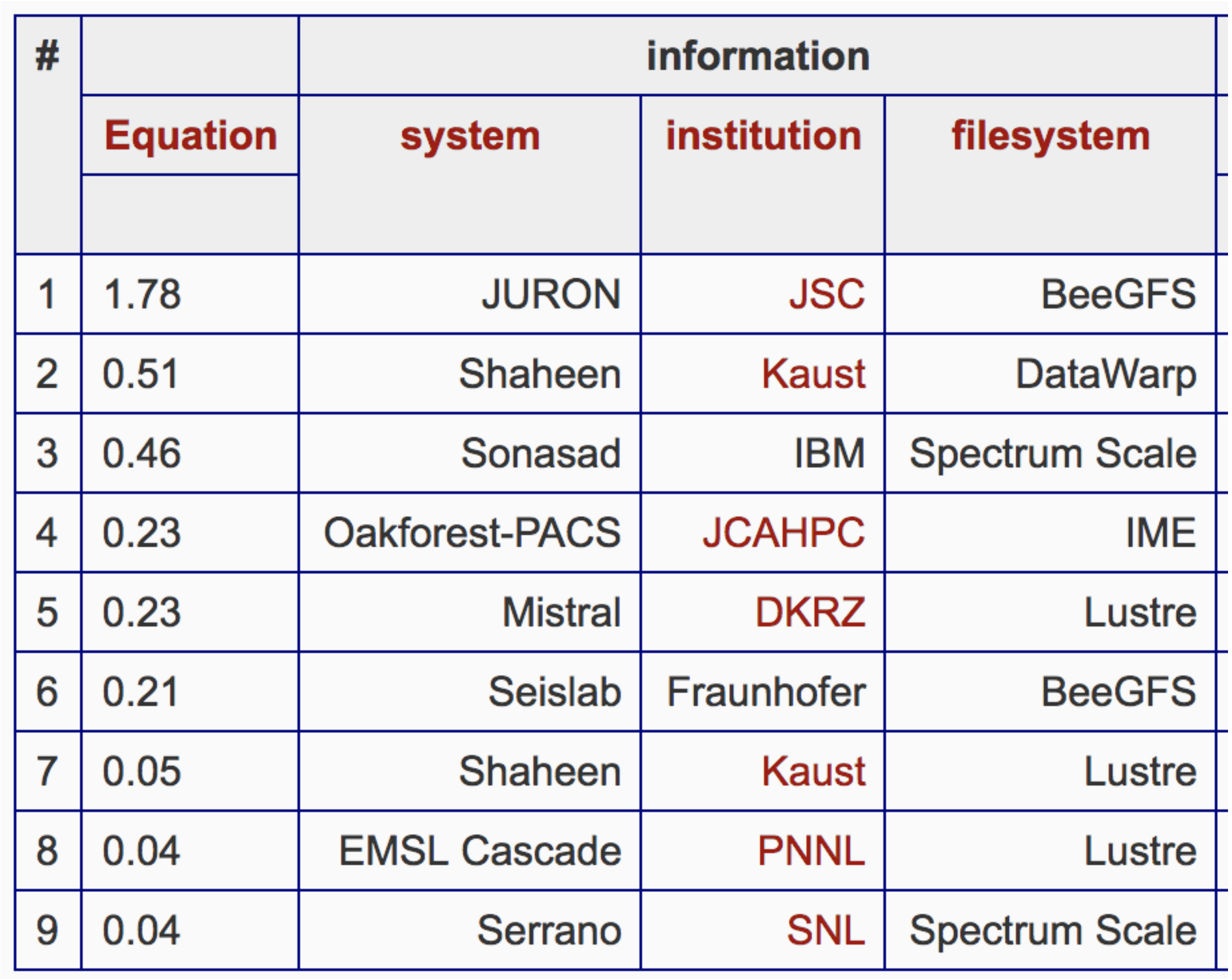
Lustre doesn’t degrade

Per-client KIOPS



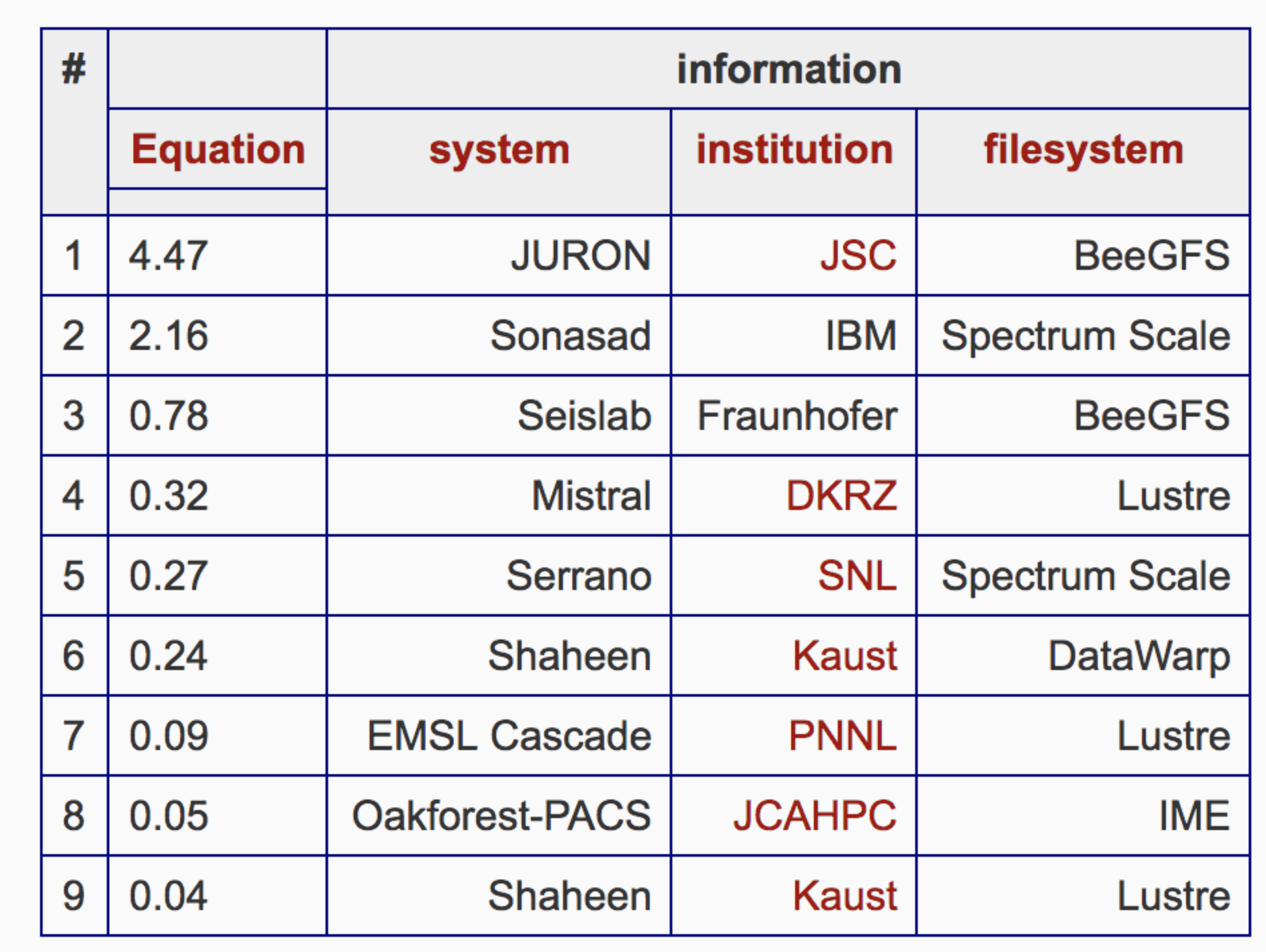
#		information		
	Equation	system	institution	filesystem
1	11.23	JURON	JSC	BeeGFS
2	10.24	Sonasad	IBM	Spectrum Scale
3	2.86	Seislab	Fraunhofer	BeeGFS
4	1.75	Serrano	SNL	Spectrum Scale
5	0.47	Mistral	DKRZ	Lustre
6	0.20	EMSL Cascade	PNNL	Lustre
7	0.11	Shaheen	Kaust	DataWarp
8	0.03	Shaheen	Kaust	Lustre
9	0.01	Oakforest-PACS	JCAHPC	IME

Per-client Bandwidth



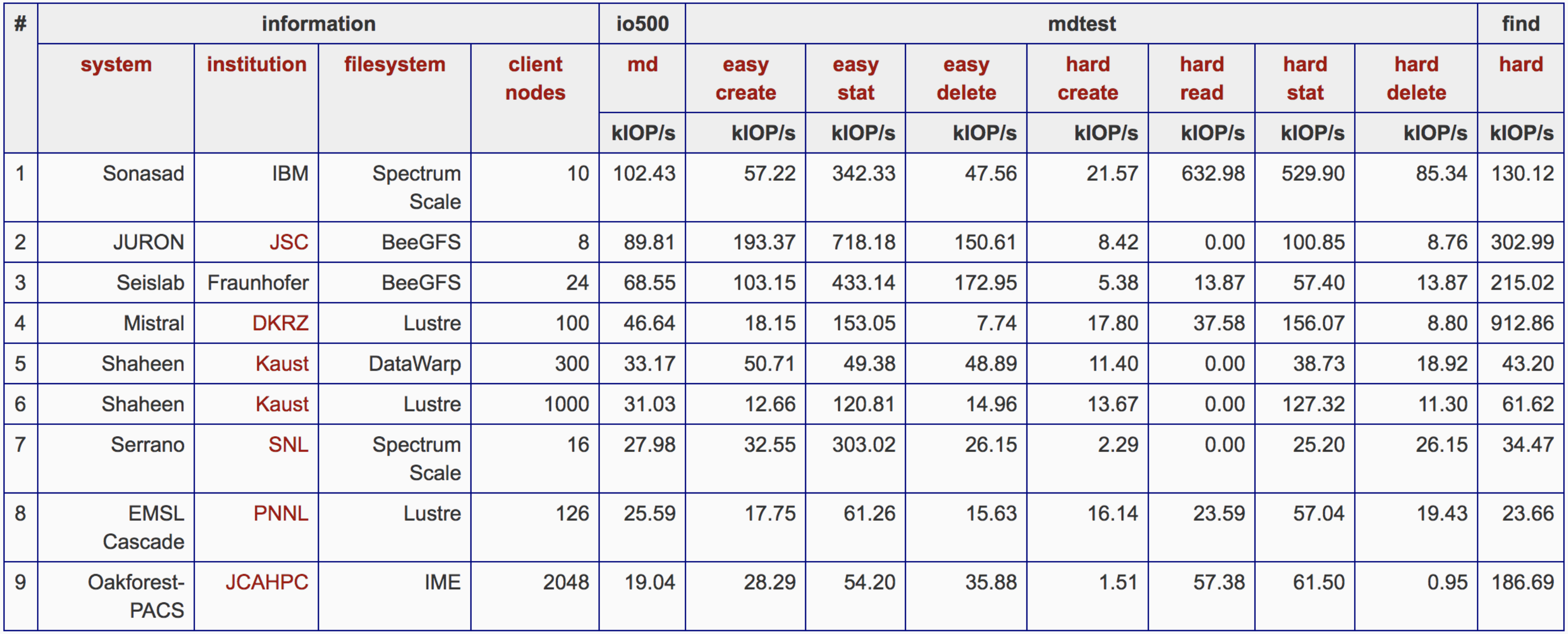
#		information		
	Equation	system	institution	filesystem
1	1.78	JURON	JSC	BeeGFS
2	0.51	Shaheen	Kaust	DataWarp
3	0.46	Sonasad	IBM	Spectrum Scale
4	0.23	Oakforest-PACS	JCAHPC	IME
5	0.23	Mistral	DKRZ	Lustre
6	0.21	Seislab	Fraunhofer	BeeGFS
7	0.05	Shaheen	Kaust	Lustre
8	0.04	EMSL Cascade	PNNL	Lustre
9	0.04	Serrano	SNL	Spectrum Scale

Per-client Score



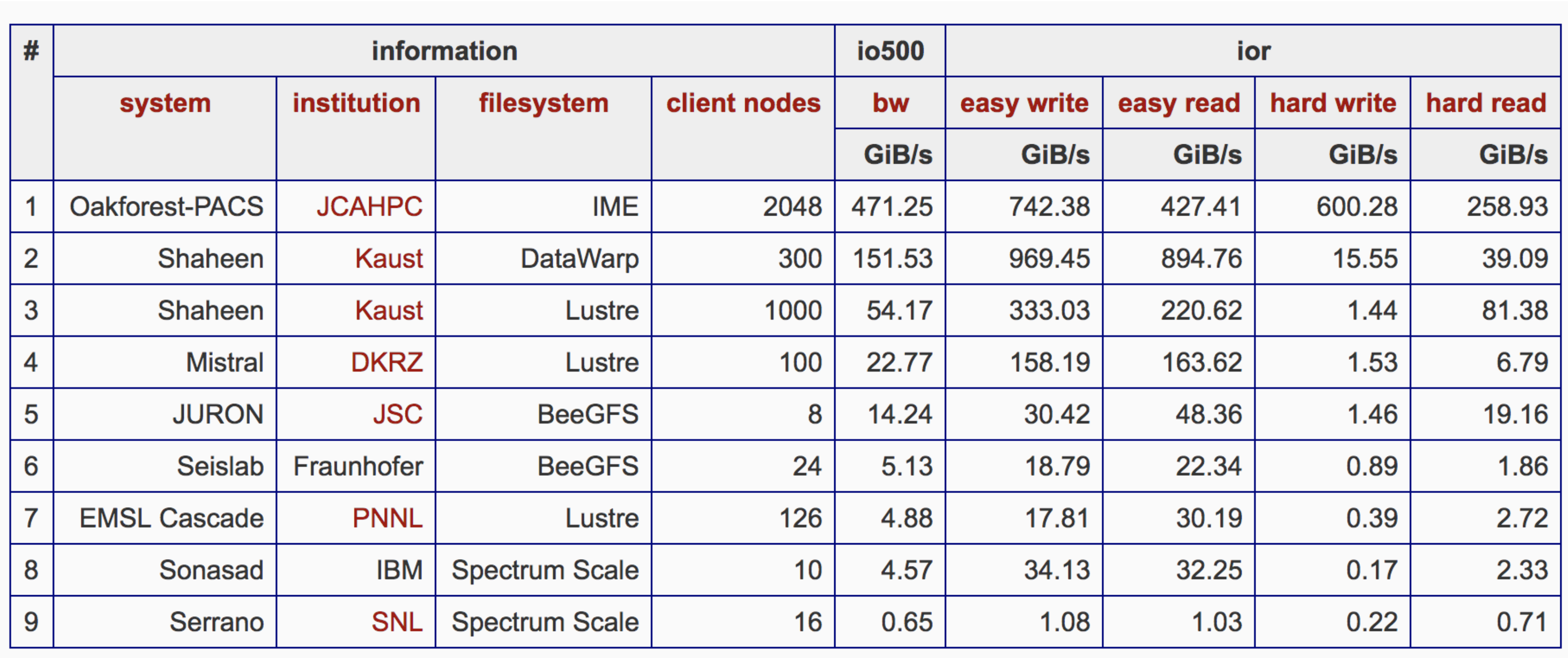
#		information		
	Equation	system	institution	filesystem
1	4.47	JURON	JSC	BeeGFS
2	2.16	Sonasad	IBM	Spectrum Scale
3	0.78	Seislab	Fraunhofer	BeeGFS
4	0.32	Mistral	DKRZ	Lustre
5	0.27	Serrano	SNL	Spectrum Scale
6	0.24	Shaheen	Kaust	DataWarp
7	0.09	EMSL Cascade	PNNL	Lustre
8	0.05	Oakforest-PACS	JCAHPC	IME
9	0.04	Shaheen	Kaust	Lustre

Highest KIOPS



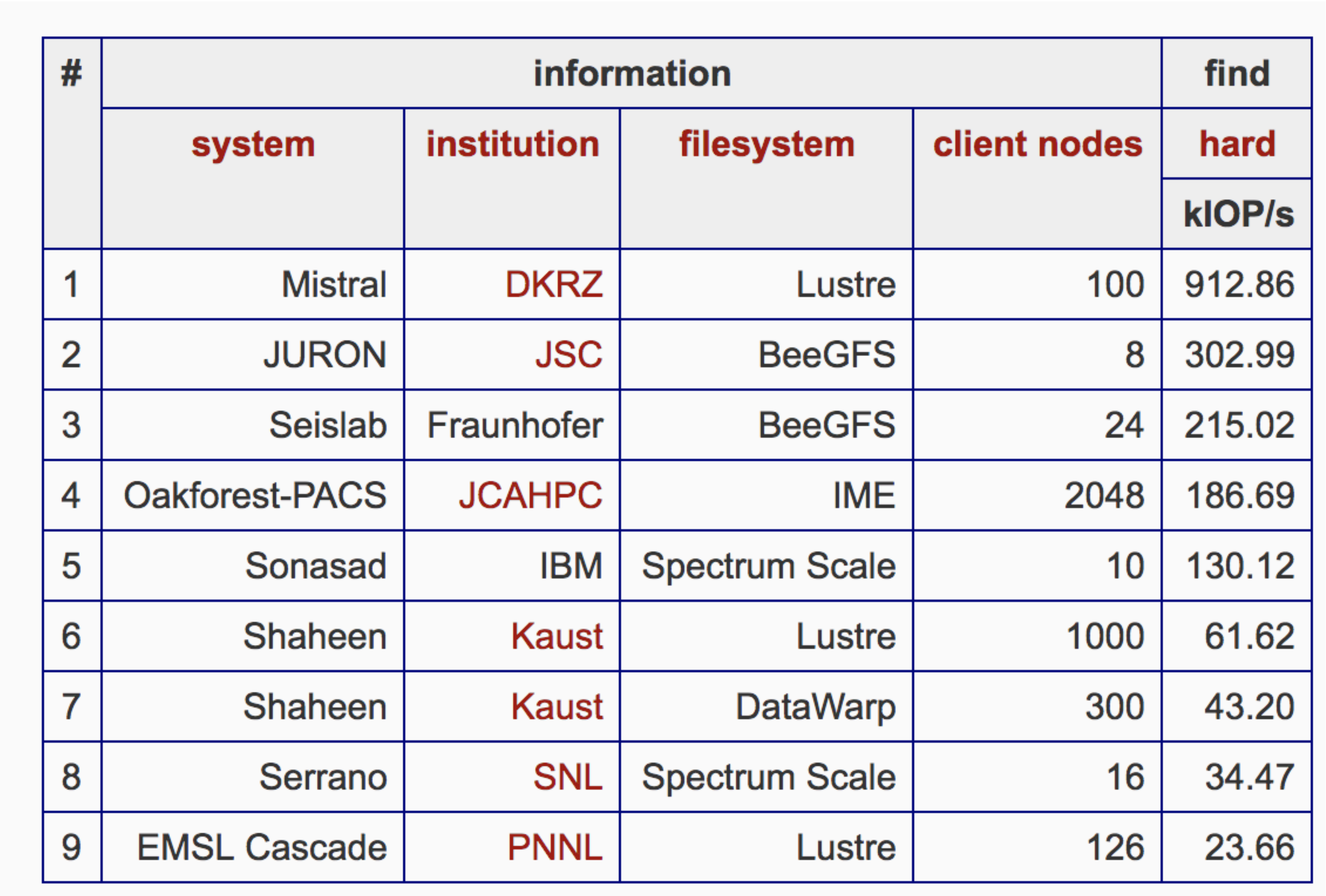
#	information				io500	mdtest							find
	system	institution	filesystem	client nodes	md	easy create	easy stat	easy delete	hard create	hard read	hard stat	hard delete	hard
					kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s
1	Sonasad	IBM	Spectrum Scale	10	102.43	57.22	342.33	47.56	21.57	632.98	529.90	85.34	130.12
2	JURON	JSC	BeeGFS	8	89.81	193.37	718.18	150.61	8.42	0.00	100.85	8.76	302.99
3	Seislab	Fraunhofer	BeeGFS	24	68.55	103.15	433.14	172.95	5.38	13.87	57.40	13.87	215.02
4	Mistral	DKRZ	Lustre	100	46.64	18.15	153.05	7.74	17.80	37.58	156.07	8.80	912.86
5	Shaheen	Kaust	DataWarp	300	33.17	50.71	49.38	48.89	11.40	0.00	38.73	18.92	43.20
6	Shaheen	Kaust	Lustre	1000	31.03	12.66	120.81	14.96	13.67	0.00	127.32	11.30	61.62
7	Serrano	SNL	Spectrum Scale	16	27.98	32.55	303.02	26.15	2.29	0.00	25.20	26.15	34.47
8	EMSL Cascade	PNNL	Lustre	126	25.59	17.75	61.26	15.63	16.14	23.59	57.04	19.43	23.66
9	Oakforest-PACS	JCAHPC	IME	2048	19.04	28.29	54.20	35.88	1.51	57.38	61.50	0.95	186.69

Highest Bandwidth



#	information				io500	ior			
	system	institution	filesystem	client nodes	bw	easy write	easy read	hard write	hard read
					GiB/s	GiB/s	GiB/s	GiB/s	GiB/s
1	Oakforest-PACS	JCAHPC	IME	2048	471.25	742.38	427.41	600.28	258.93
2	Shaheen	Kaust	DataWarp	300	151.53	969.45	894.76	15.55	39.09
3	Shaheen	Kaust	Lustre	1000	54.17	333.03	220.62	1.44	81.38
4	Mistral	DKRZ	Lustre	100	22.77	158.19	163.62	1.53	6.79
5	JURON	JSC	BeeGFS	8	14.24	30.42	48.36	1.46	19.16
6	Seislab	Fraunhofer	BeeGFS	24	5.13	18.79	22.34	0.89	1.86
7	EMSL Cascade	PNNL	Lustre	126	4.88	17.81	30.19	0.39	2.72
8	Sonasad	IBM	Spectrum Scale	10	4.57	34.13	32.25	0.17	2.33
9	Serrano	SNL	Spectrum Scale	16	0.65	1.08	1.03	0.22	0.71

Fastest “Find”



#	information				find
	system	institution	filesystem	client nodes	hard
					kIOP/s
1	Mistral	DKRZ	Lustre	100	912.86
2	JURON	JSC	BeeGFS	8	302.99
3	Seislab	Fraunhofer	BeeGFS	24	215.02
4	Oakforest-PACS	JCAHPC	IME	2048	186.69
5	Sonasad	IBM	Spectrum Scale	10	130.12
6	Shaheen	Kaust	Lustre	1000	61.62
7	Shaheen	Kaust	DataWarp	300	43.20
8	Serrano	SNL	Spectrum Scale	16	34.47
9	EMSL Cascade	PNNL	Lustre	126	23.66

Questions?

- Visit the site
- http://io500.org