IO-500

http://io500.org

Why an IO-500

- Honesty
  - Sick of sites only advertising hero numbers
- Expectations
  - Users only see hero numbers and don't know what realistic IO performance is
- Community
  - Collect a repository of results along with info on how tuning was done
- More balanced systems
  - Make procurement of systems pay more attention to storage
- Better storage
  - Force parallel file system developers to focus on the anti-hero workloads
- Easier RFP writing
  - We have already seen an RFP specifying the mdtest_hard parameters.

What is IO-500

- Bandwidth
  - IOR easy run: user’s choice
  - IOR hard run: single-shared file, small unaligned, POSIX
- Metadata
  - mdtest easy run: user’s choice
  - mdtest hard run: single-shared directory, 3901 byte files, POSIX
  - “find” functionality
    - Of all the files created in the last ~20 minutes, all those that were created
      - in the last ~10 minutes
      - with size 3901
      - matching string “01”

How the IO500

```
>git clone https://github.com/VI4IO/io-500-dev
>cd io-500-dev
>./utilities/prepare.sh
>./io500.sh
># tune (write/create phases must last for 5 minutes)
># email submit@io500.org
```

Making it easier and improving the existing tools

- Made “stonewall” better in IOR
- Fixed read verification in IOR
- Merged IOR and mdtest
- Added an MPI “find”
- Planning on adding ”stonewall” to mdtest
- Planning on adding read verification to mdtest

The “find” command is challenging

- Just running ./find serially can take a long time
- We added an MPI find
- We are hoping for lots of community innovation here

- GPFS has a nice solution: *mmfind*

mmfind

- Description:
  - A highly efficient file system traversal tool, designed to serve as a drop-in replacement for the 'find' command as used against GPFS FSes.
  - It is a wrapper around the GPFS mmapplypolicy command.
  - It is based on 'Posix find', with support for some GNU extensions.
  - It also has some GPFS-specific features.
  - It will be slower than 'find' in small-enough FSes, but at scale mmfind will be significantly faster.

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

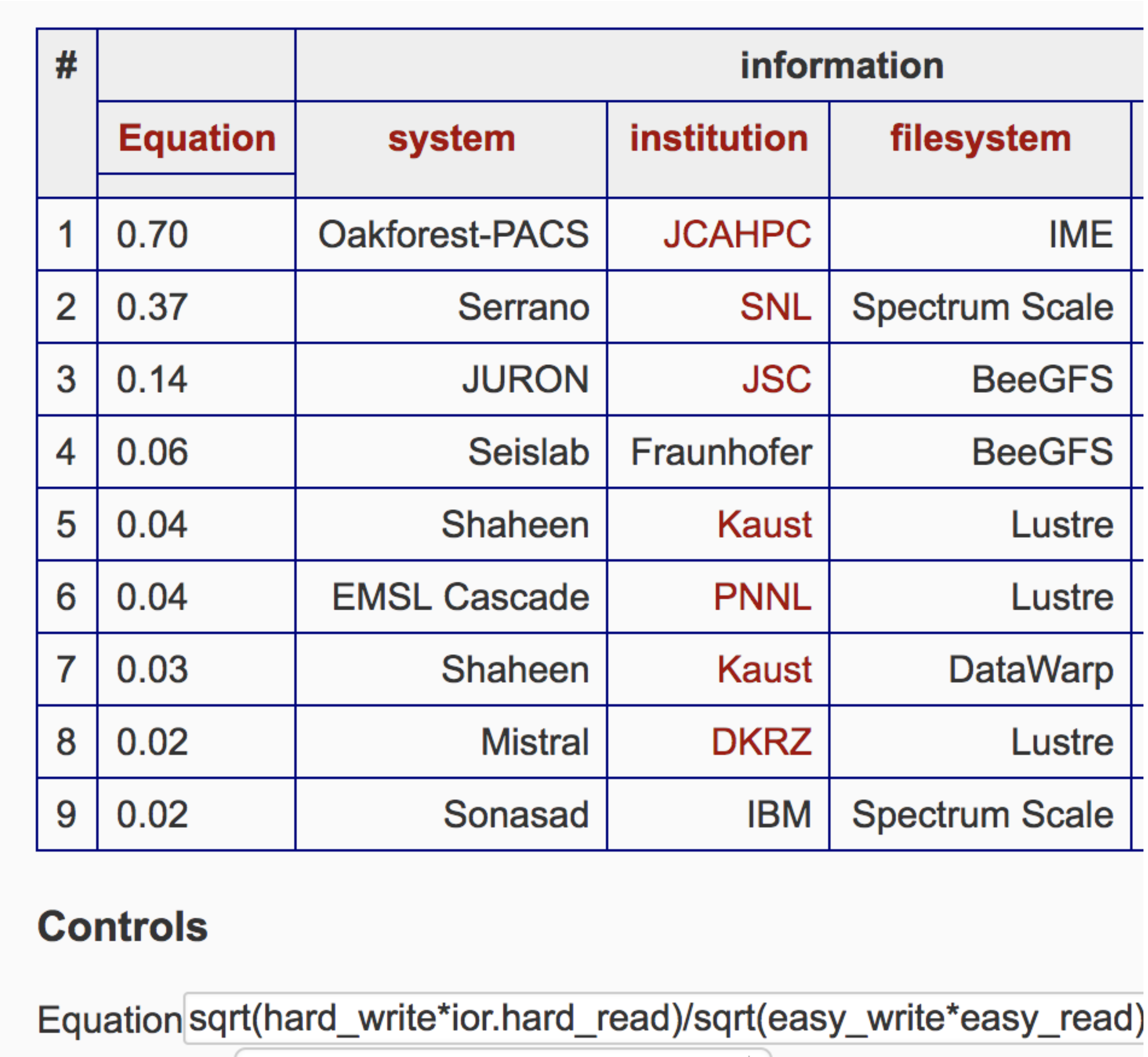
- bandwidth
  - geo_mean of the IOR scores
- iops
  - geo_mean of mdtest scores and “find”
- total
  - sq_root(bandwidth*iops)

Thanks

- To the entire community and mailing list
  - Please join mailing list. Info at http://io500.org
- To all submitters
- To all testers
- To all who contributed to the github repo’s

And now some results

Least degradation from IOR easy to hard

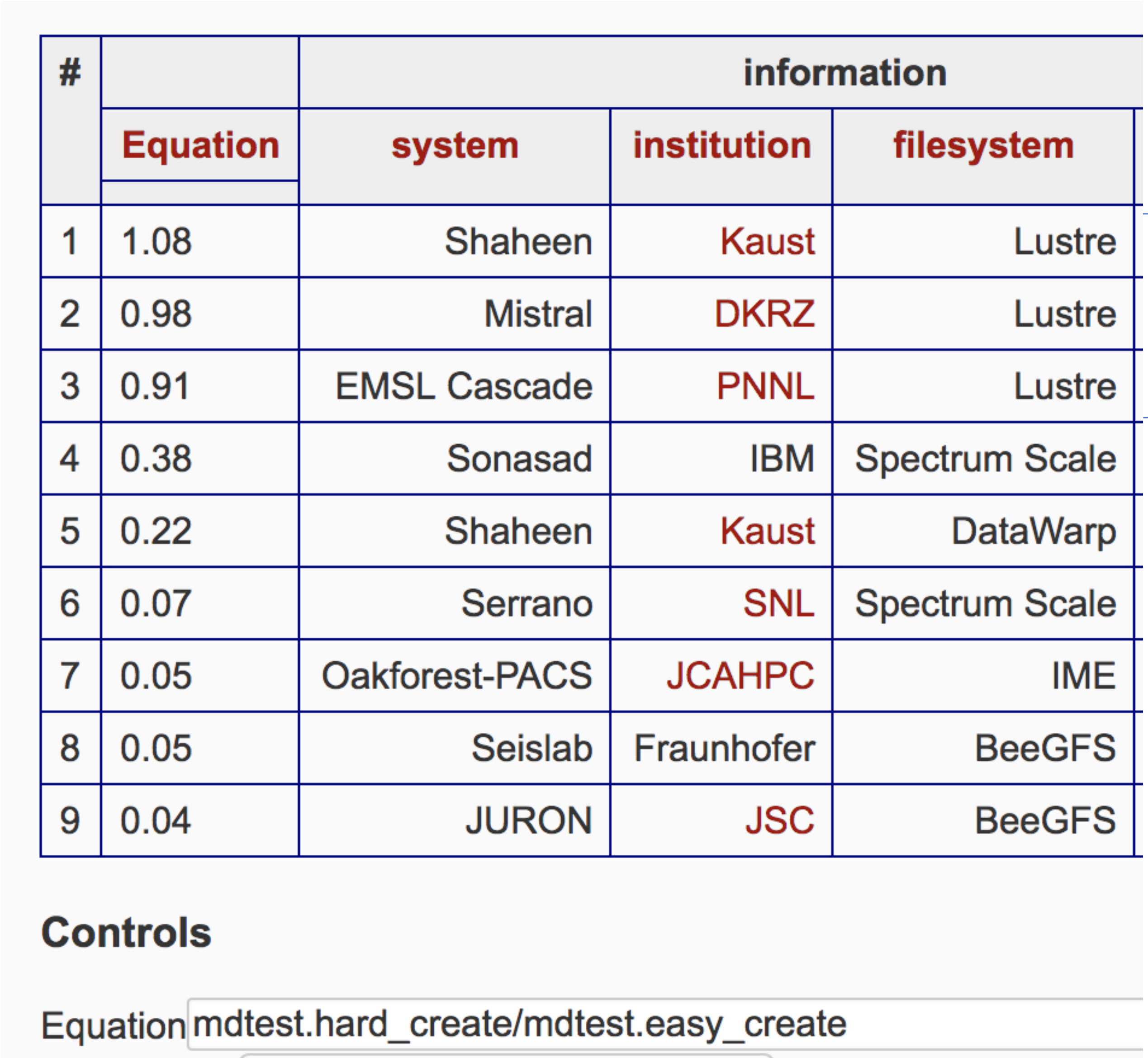


#		information		
	Equation	system	institution	filesystem
1	0.70	Oakforest-PACS	JCAHPC	IME
2	0.37	Serrano	SNL	Spectrum Scale
3	0.14	JURON	JSC	BeeGFS
4	0.06	Seislab	Fraunhofer	BeeGFS
5	0.04	Shaheen	Kaust	Lustre
6	0.04	EMSL Cascade	PNNL	Lustre
7	0.03	Shaheen	Kaust	DataWarp
8	0.02	Mistral	DKRZ	Lustre
9	0.02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

Controls

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

Degradation for creates in shared directory



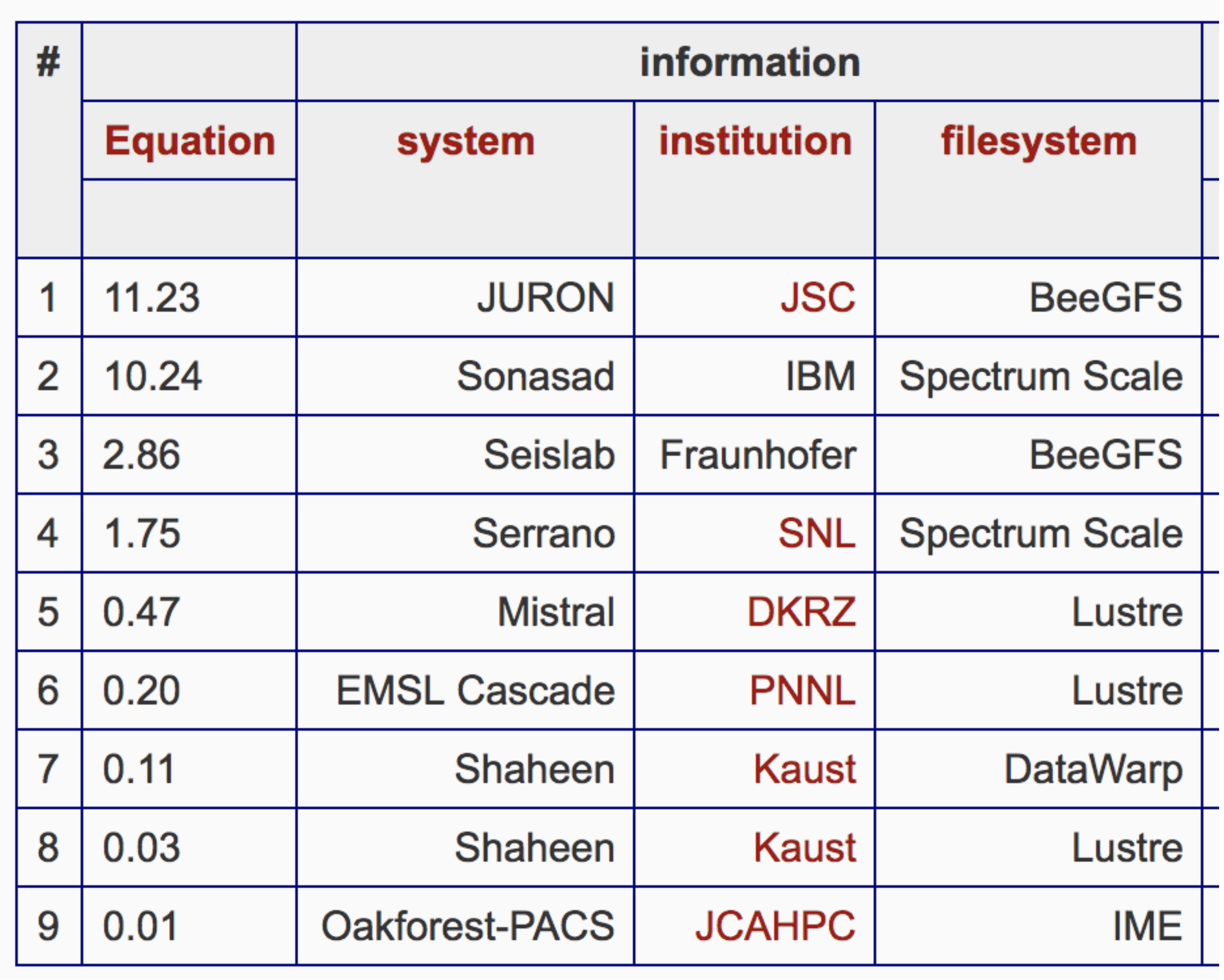
#		information		
	Equation	system	institution	filesystem
1	1.08	Shaheen	Kaust	Lustre
2	0.98	Mistral	DKRZ	Lustre
3	0.91	EMSL Cascade	PNNL	Lustre
4	0.38	Sonasad	IBM	Spectrum Scale
5	0.22	Shaheen	Kaust	DataWarp
6	0.07	Serrano	SNL	Spectrum Scale
7	0.05	Oakforest-PACS	JCAHPC	IME
8	0.05	Seislab	Fraunhofer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

Controls

Equation mdtest.hard_create/mdtest.easy_create

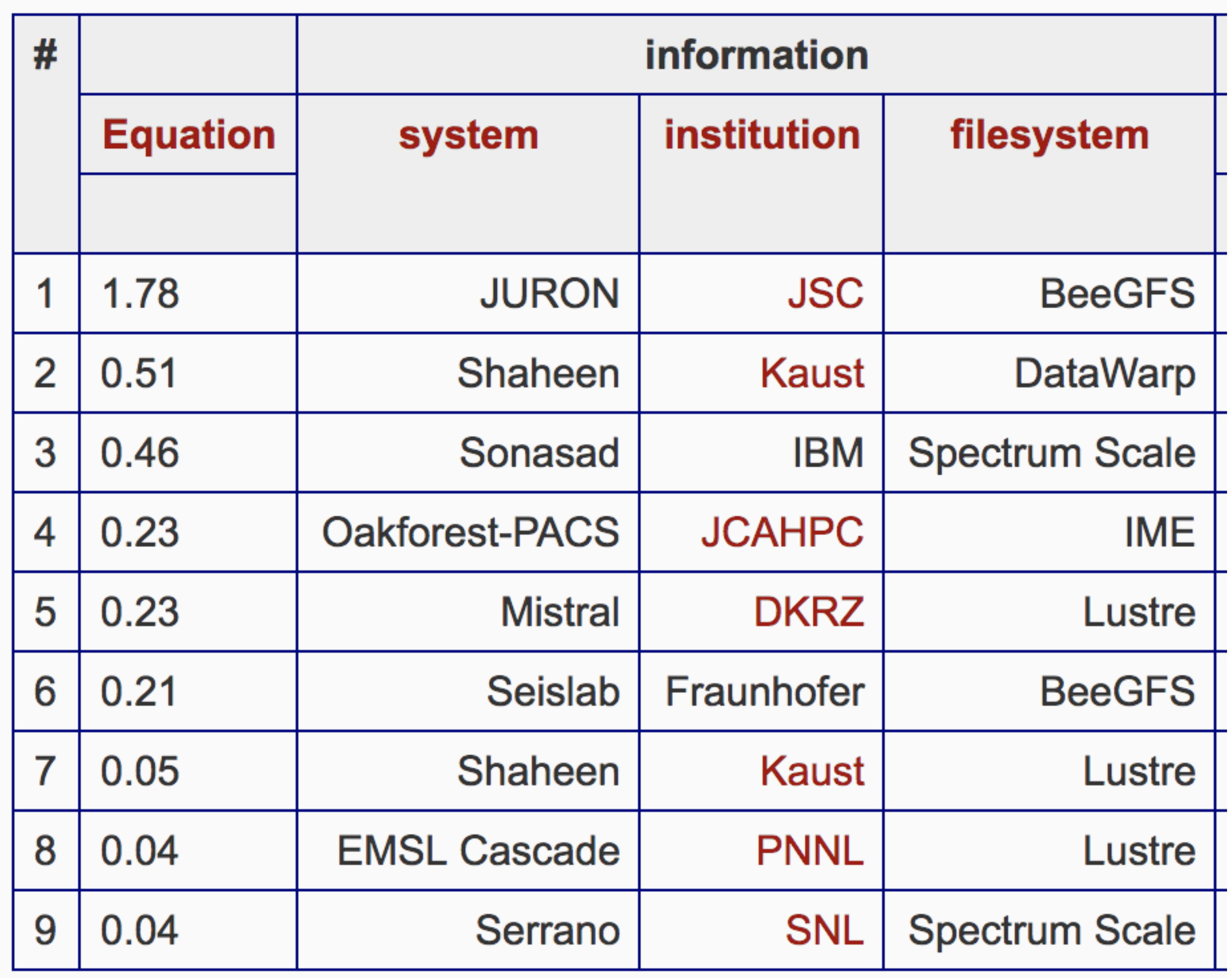
Lustre doesn’t degrade

Per-client KIOPS



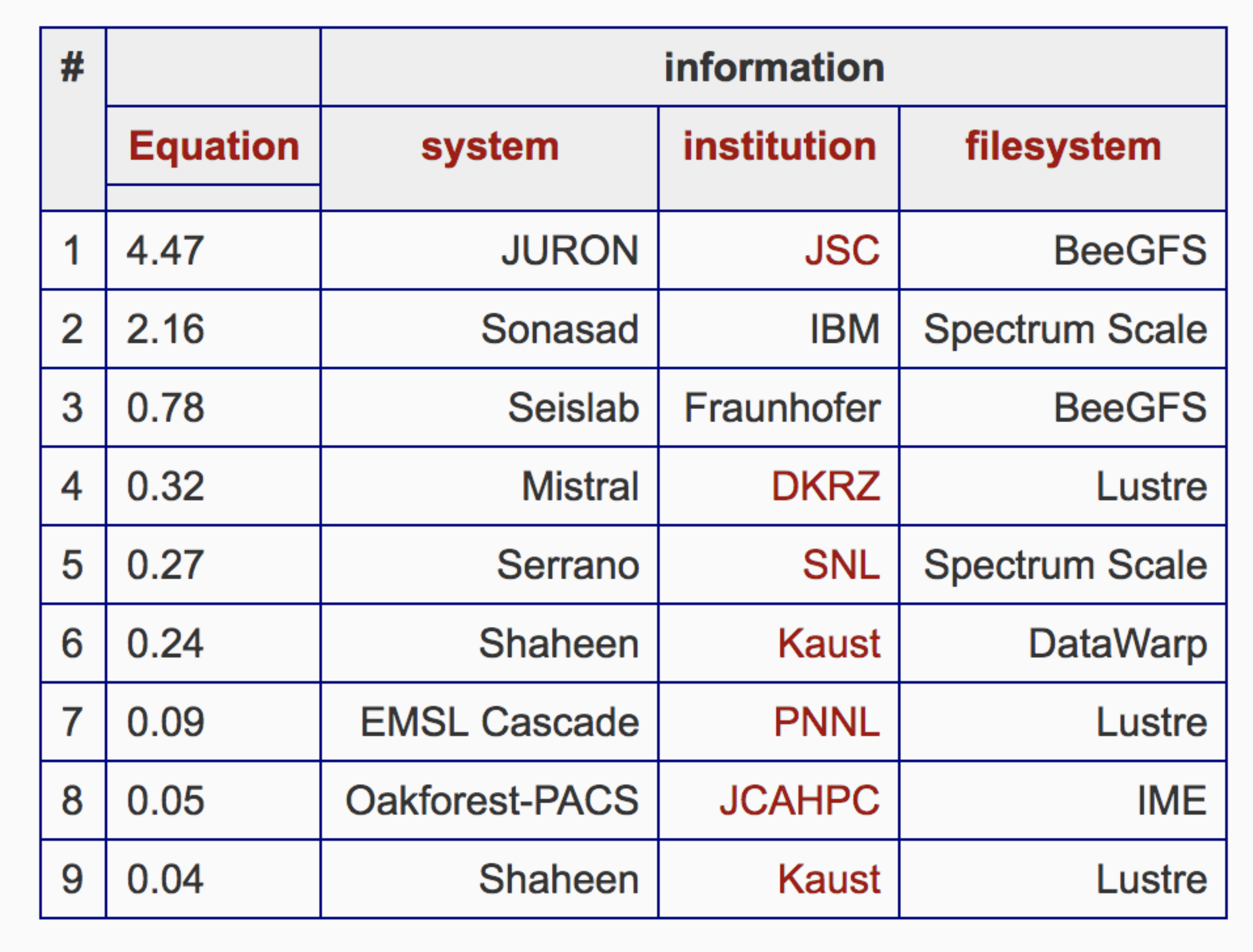
#		information		
	Equation	system	institution	filesystem
1	11.23	JURON	JSC	BeeGFS
2	10.24	Sonasad	IBM	Spectrum Scale
3	2.86	Seislab	Fraunhofer	BeeGFS
4	1.75	Serrano	SNL	Spectrum Scale
5	0.47	Mistral	DKRZ	Lustre
6	0.20	EMSL Cascade	PNNL	Lustre
7	0.11	Shaheen	Kaust	DataWarp
8	0.03	Shaheen	Kaust	Lustre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

Per-client Bandwidth



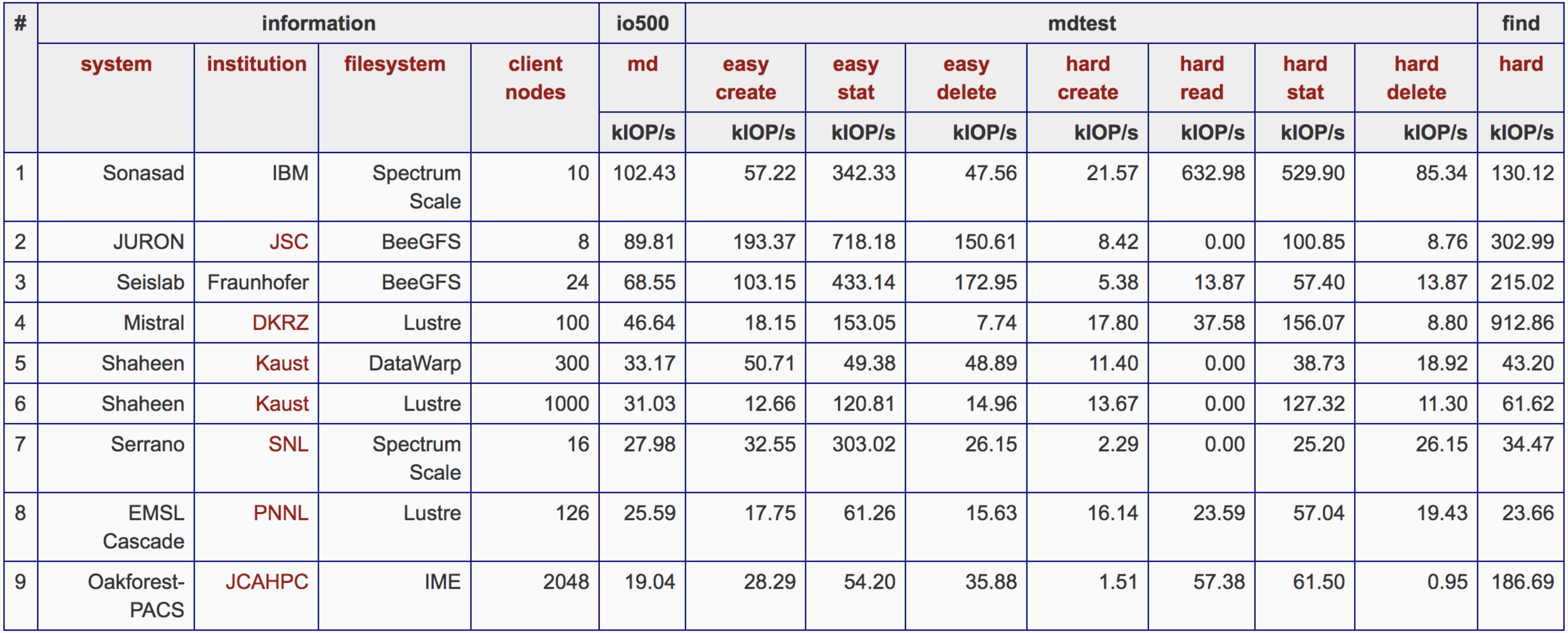
#		information		
	Equation	system	institution	filesystem
1	1.78	JURON	JSC	BeeGFS
2	0.51	Shaheen	Kaust	DataWarp
3	0.46	Sonasad	IBM	Spectrum Scale
4	0.23	Oakforest-PACS	JCAHPC	IME
5	0.23	Mistral	DKRZ	Lustre
6	0.21	Seislab	Fraunhofer	BeeGFS
7	0.05	Shaheen	Kaust	Lustre
8	0.04	EMSL Cascade	PNNL	Lustre
9	0.04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

Per-client Score



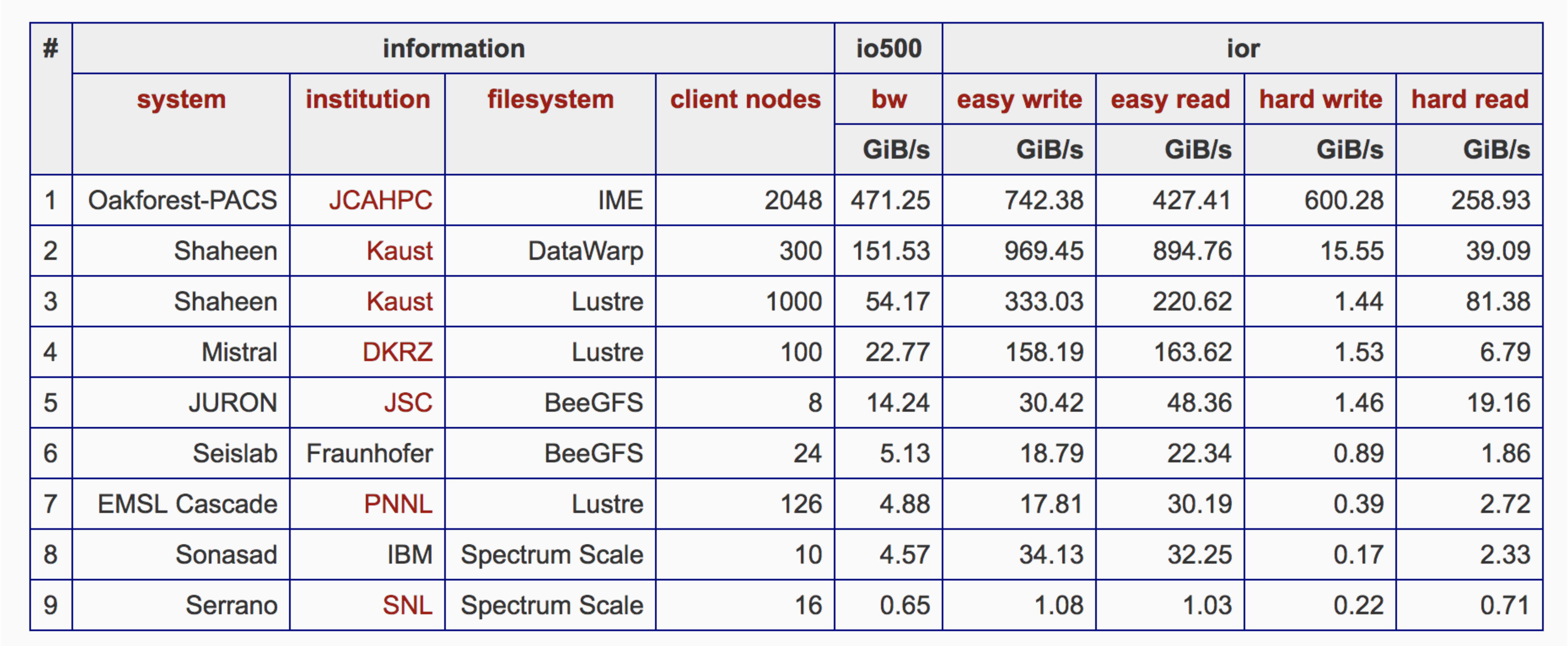
#		information		
	Equation	system	institution	filesystem
1	4.47	JURON	JSC	BeeGFS
2	2.16	Sonasad	IBM	Spectrum Scale
3	0.78	Seislab	Fraunhofer	BeeGFS
4	0.32	Mistral	DKRZ	Lustre
5	0.27	Serrano	SNL	Spectrum Scale
6	0.24	Shaheen	Kaust	DataWarp
7	0.09	EMSL Cascade	PNNL	Lustre
8	0.05	Oakforest-PACS	JCAHPC	IME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

Highest KIOPS



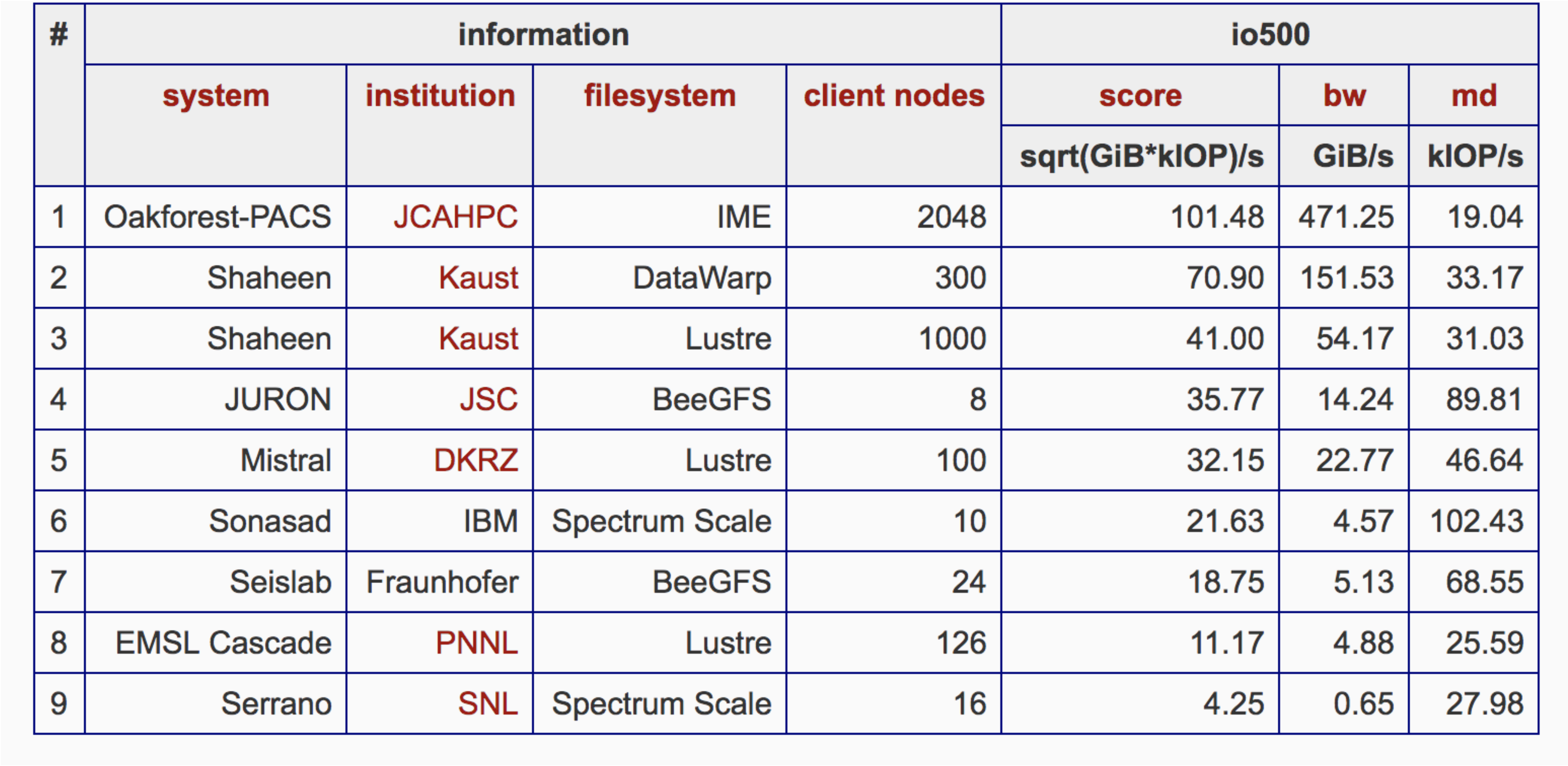
#	information				io500	mdtest							find
	system	institution	filesystem	client nodes	md	easy create	easy stat	easy delete	hard create	hard read	hard stat	hard delete	hard
					kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s
1	Sonasad	IBM	Spectrum Scale	10	102.43	57.22	342.33	47.56	21.57	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

First Annual IO500 Winner: Oakforest-PACS

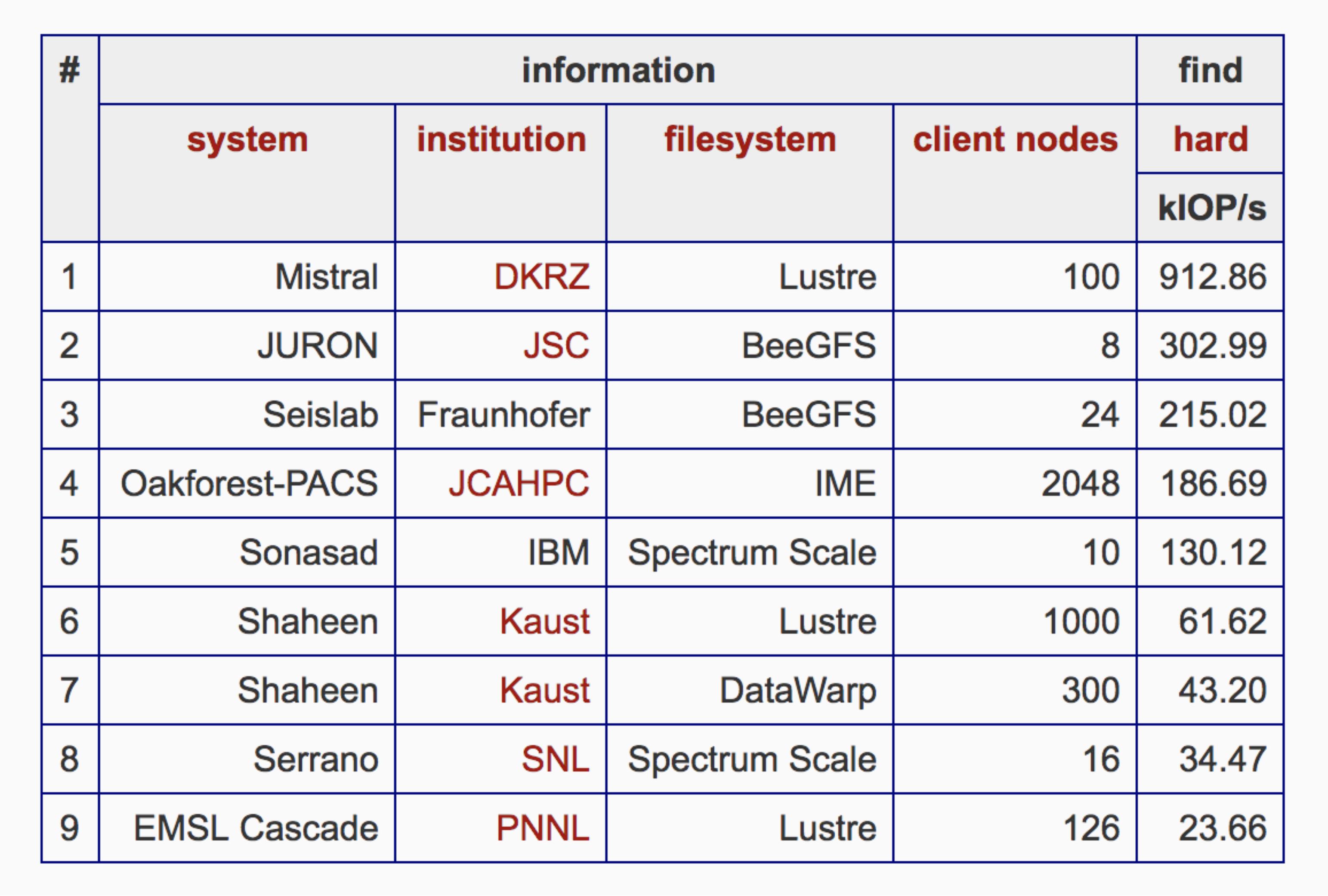


#	information				io500		
	system	institution	filesystem	client nodes	score	bw	md
					sqrt(GiB*kIOP)/s	GiB/s	kIOP/s
1	Oakforest-PACS	JCAHPC	IME	2048	101.48	471.25	19.04
2	Shaheen	Kaust	DataWarp	300	70.90	151.53	33.17
3	Shaheen	Kaust	Lustre	1000	41.00	54.17	31.03
4	JURON	JSC	BeeGFS	8	35.77	14.24	89.81
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

Congrats to JCAHPC!

- See you next year at ISC’18
- Please get involved at http://io500.org

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