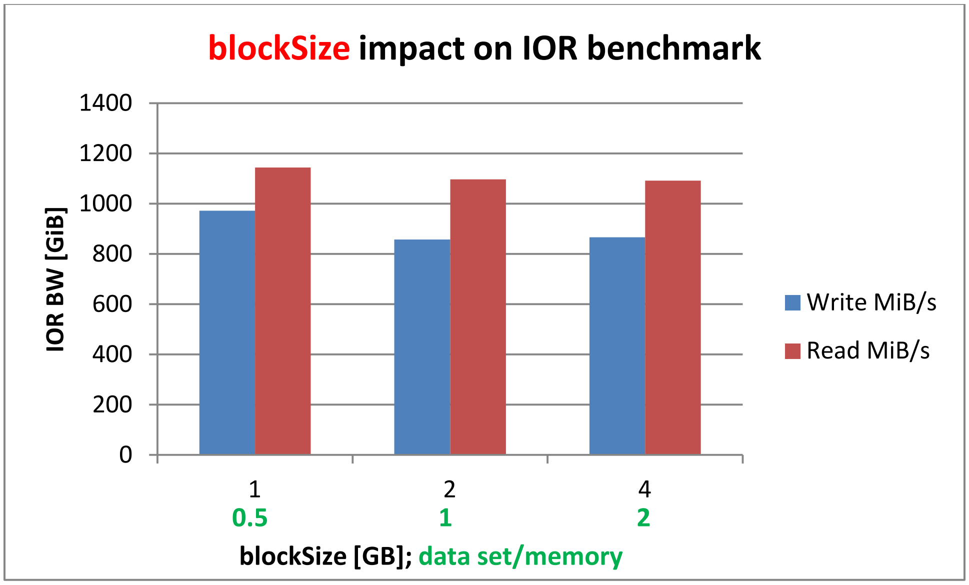
blockSize impact on IOR performance

We limited our tests to only POSIX case as the MPIIO case behaves almost identical and in some cases better under caching effects than the POSIX. So POSIX is the worst case.

blocksize			
cached	Write	Read	set/mem
new set	MiB/s	MiB/s	
1	972	1144	0.5
2	857	1097	1
4	866	1092	2

Note: These tests were performed for different blockSize values of, 1GB, 2GB and 4GB, using each time new data sets both for write as well as for read. We ran separate write tests and read tests reusing same data sets that were written but flashing the caches of the clients and servers in order to eliminate any caching effects for data blocks residing in the caches. We used cached IO to allow the Lustre client caching for pre-fetch but we ensured that the cache did not contain blocks from the test data set. We used only blocksize values that will demonstrate data set to aggregate memory ratios of 0.5x (data set smaller than aggregate memory), 1x (data set equal to memory) and data set 2x aggregate memory size.



Note: The results in the chart show that the performance of both write and read is not dependent of the ratio of data set to memory either for smaller as well as larger caches if we use always new data sets and separate the write benchmark from read benchmark and do not reuse old data set more than once. In order to achieve this we created 10 data sets and ensured to roll them such that no cache effects are present. There is some small advantage for the read performance when the data set is smaller than the memory but we believe it has to do with the caching of the Lustre client when there is spare memory for MD caching, maybe.

-b 1g; -t 4m	Write	Read	set/mem	Notes
	MiB/s	MiB/s		
o_direct	972	1480 (12Gbit)	0.5	>10GbE BW
cached	793	32828 (400Gbit)	0.5	>>10GbE BW
new	909	1144	0.5	TRUE

Note: In this test series we compared the results of different caching method on the write and read performance for the case when the data set is smaller than the memory. The main conclusion is that o_direct option does not guarantee correct read measurement for the benchmark in the cases when the data is cached and re-used. This will indicate that the o_direct (-B option) is not enough to guarantee correct results for read if the data set is cached in any of the caches: client, OSS, OST and block storage. Another interesting observation is that for the POSIX case is that the write performance with o_direct for smaller data sets is slightly higher than the cached case when using new data sets and even sensitive higher that the re-write tests when the data set is in the cache of the host and the file is overwritten.

As for the read performance it is obvious that when all the data set is in the cache the BW will be bogus for the FS and will mostly show cache "speed" BW. But it is important to notice that the read BW performance is correct, more than o_direct case, when using new data sets even if they are smaller than the aggregate memory.

Based on this observation one can conclude that as long as one uses new data sets that have no blocks in any memory of the system the read BW measurements are correct and this will be our recommendation for IOR read BW benchmark: Create enough data sets with aggregate size 10x larger than the aggregate memory of the system and run read tests with the oldest data set touched for either write or read. Although creating 10 data sets can take a long time, considering the case when using much larger new data sets that can take longer test time due to the larger size of the data set, the test time will be shorter to get accurate measurements.