LDMS-Darshan analysis

Ana Luísa Veroneze Solórzano
Devesh Tiwari
Rohan Basu Roy
Sara Walton
Benjamin Schwaller
Jim M. Brandt

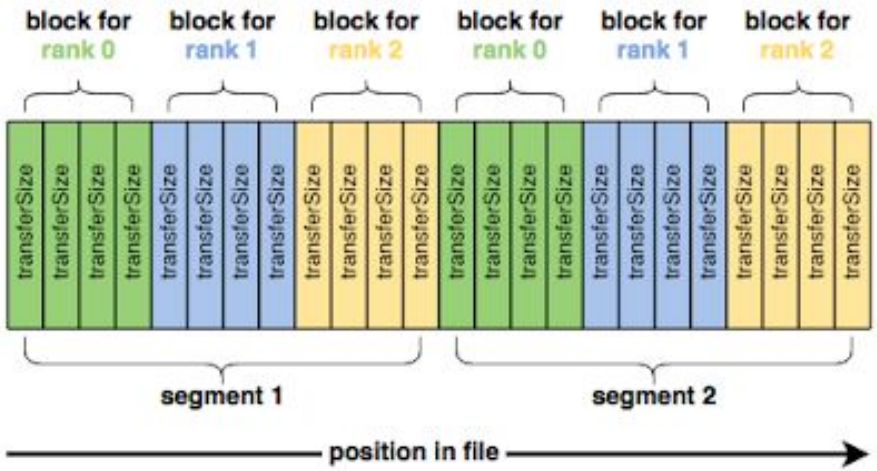
IOR - I/O Benchmark

Available in: https://github.com/hpc/ior

Interface for POSIX, MPI-IO, HDF5, HDFS, S3, NCMPI, IME, MMAP, RADOS

User can configure: operation (-w/-r), block size (-b), segments(-s), transfer size (-t), and tasks (-i)

Each rank deals with 1 block in each segment.

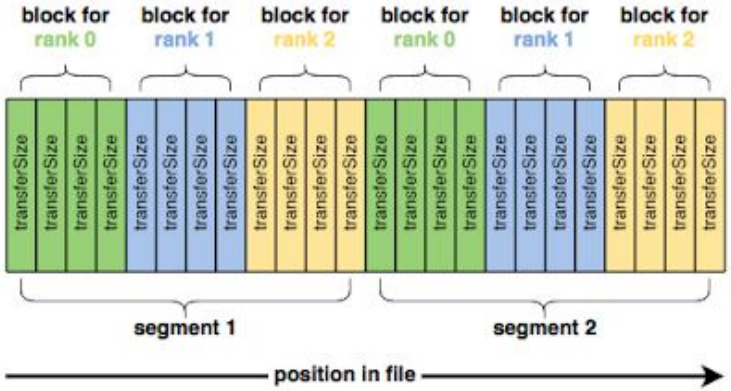


From: https://glennklockwood.blogspot.com/2016/07/basics-of-io-benchmarking.html

IOR - Execution

$ ./ior -i 6 -b 144k -t 24k -s 1024 -F -C -e -k -o /pscratch/user/iorTest/darshan

- -i: repeat the same test 6 times
- -b: blocksize of 144k
- -t: transfersize of 24k
- -s: 1024 segments
- -F: 1 file per process
- -C: forces each MPI process to read the data written by its neighboring node
- -e: flushes dirty pages after writes
- -k: keep test file(s) on program exit
- -o: Location of test file(s)



1024 segments * 32 ranks * 144K block size = 4GB per iteration

Experiments for 1 node and 32 ranks.

IOR - Results - Darshan DXT

2,097,024 I/O operations captured

- All POSIX
- 32,768 segments
- 1,048,448 reads
- 1,048,576 writes

Time ranges:

- Slurm: *2023-10-11 16:19:55 MDT - 2023-10-11 16:20:04 - 13.021s*
- DXT: 7.526 s

IOR - Results - Darshan DXT

access	bw(MiB/s)	IOPS	Latency(s)	block(KiB)	xfer(KiB)	open(s)	wr/rd(s)	close(s)	total(s)	iter
write	5658	242401	0.107624	144.00	24.00	0.039427	0.811087	0.287062	0.814471	0
read	20007	859087	0.031528	144.00	24.00	0.002908	0.228857	0.072122	0.230315	0
write	6944	297744	0.085053	144.00	24.00	0.038066	0.660325	0.209614	0.663636	1
read	19870	852512	0.026567	144.00	24.00	0.003003	0.230622	0.073473	0.231904	1
write	6839	293374	0.089394	144.00	24.00	0.035525	0.670162	0.209300	0.673812	2
read	19862	853452	0.026627	144.00	24.00	0.003182	0.230368	0.074137	0.231999	2
write	6257	268041	0.088952	144.00	24.00	0.035731	0.733499	0.269219	0.736457	3
read	19954	856634	0.026774	144.00	24.00	0.002979	0.229512	0.069547	0.230929	3
write	6928	296961	0.092403	144.00	24.00	0.033187	0.662066	0.210020	0.665098	4
read	19720	845761	0.028210	144.00	24.00	0.002773	0.232463	0.078662	0.233675	4
write	6942	297413	0.105923	144.00	24.00	0.031038	0.661061	0.211552	0.663803	5
read	20248	868886	0.027292	144.00	24.00	0.002981	0.226276	0.066069	0.227582	5

Summary of all tests:

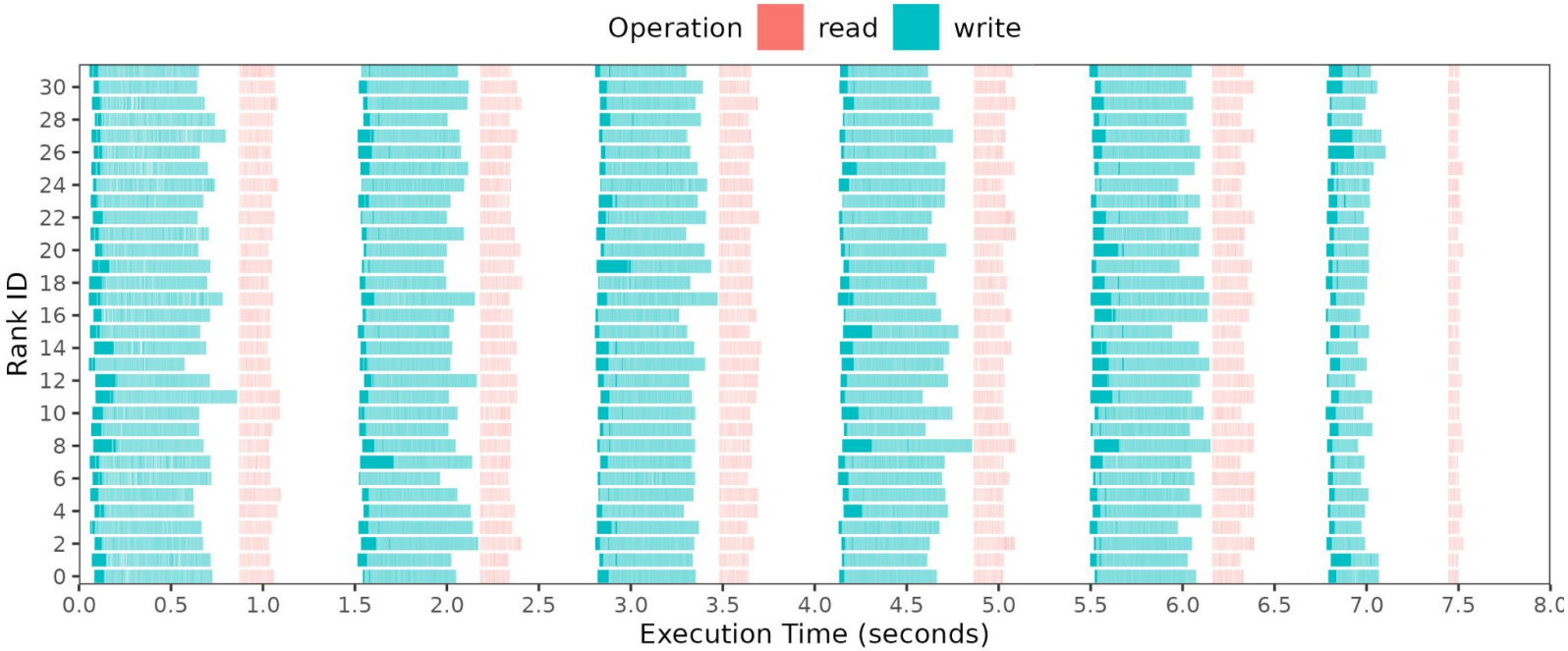
Operation	Max(MiB)	Min(MiB)	Mean(MiB)	StdDev	Max(OPs)	Min(OPs)	Mean(OPs)	StdDev	Mean(s)	Stonew
write	6943.57	5657.66	6594.51	483.93	296258.81	241393.49	281365.55	20647.62	0.70288	
read	20247.65	19719.70	19943.56	162.67	863899.69	841373.69	850925.17	6940.59	0.23107	

IOR - Results - Darshan DXT

We can identify the 6 iterations - repetitions of the same operations

IOR performs writes followed by reads

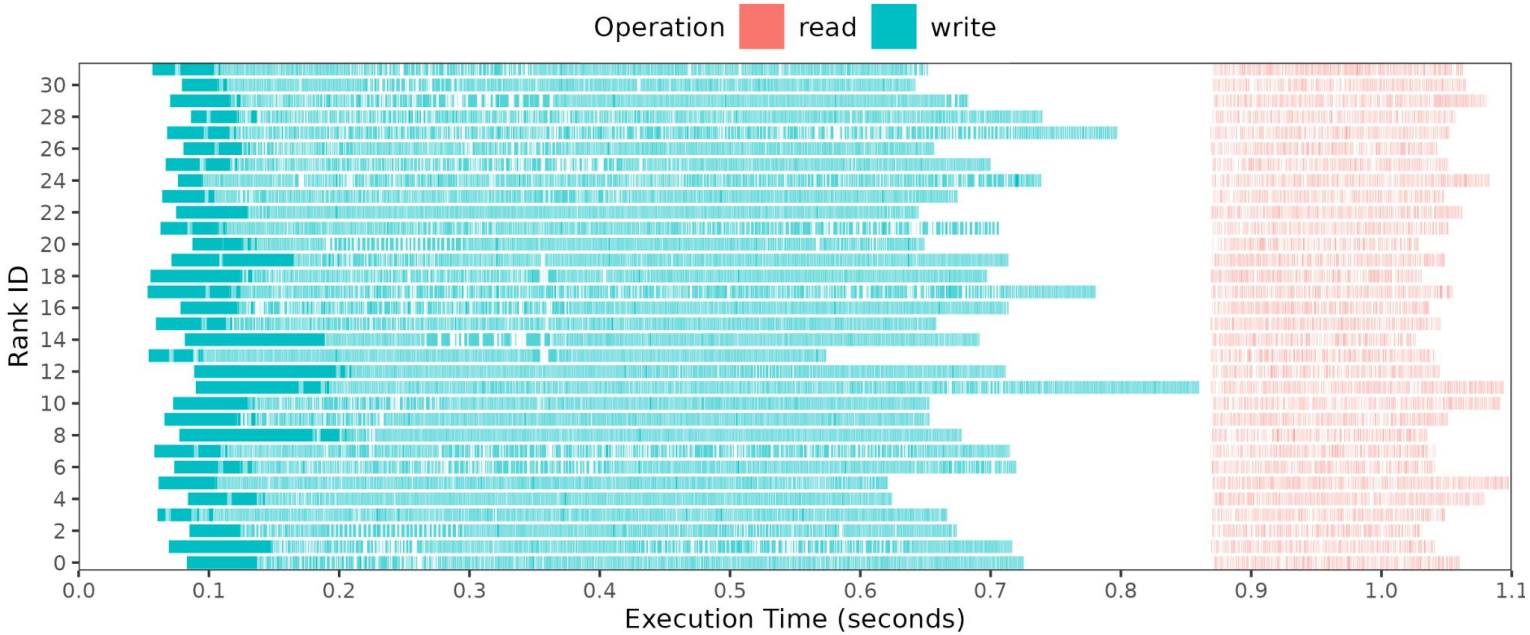
Synchronization point before reading but not before writing



IOR - Results - Darshan DXT

Synchronization point before reading but not before writing

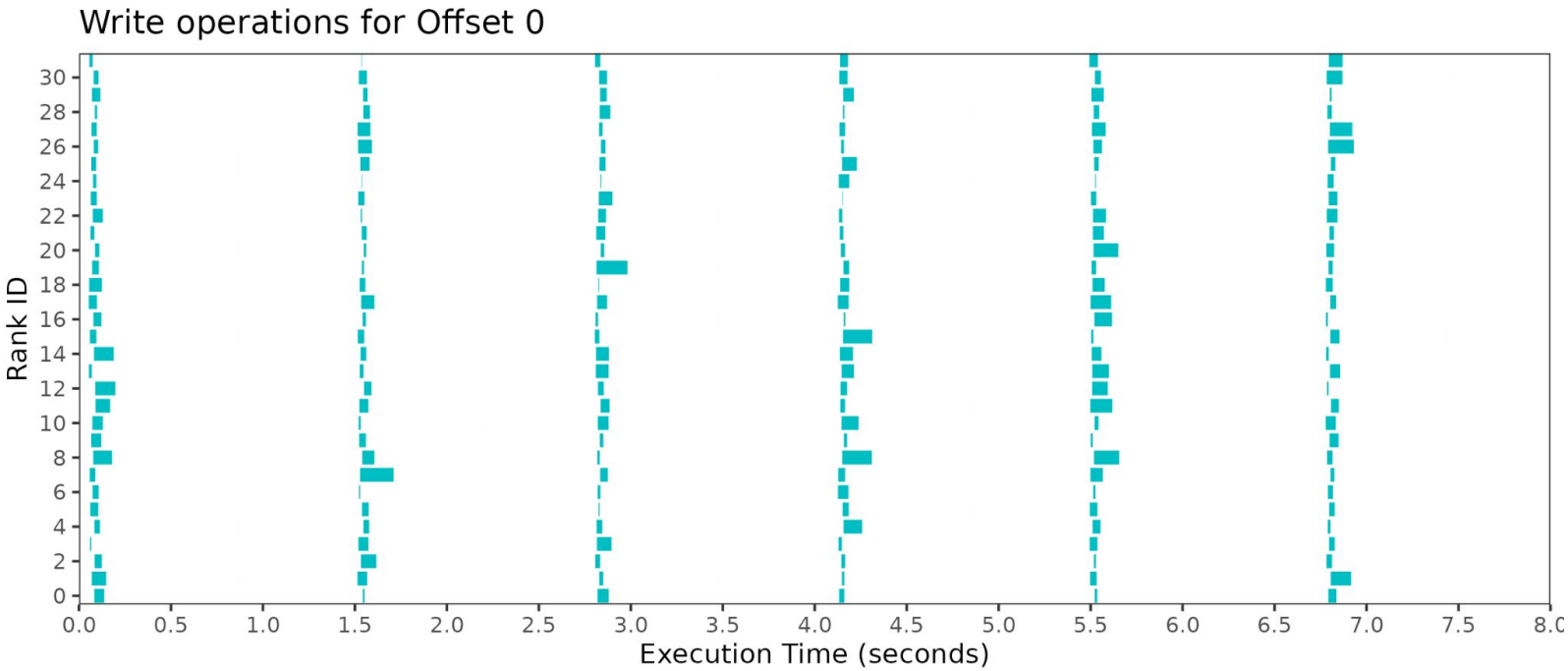
Longer writes at the beginning of each iteration



IOR - Results - Darshan DXT

Longer writes are for offset 0

Each one is writing 24KB, but they are taking different durations to process it



IOR - Results - Darshan DXT

Just capturing the first iteration!

- 339,052 POSIX and 71 STDIO (opens and closes)
- Same number of offsets and blocks sizes

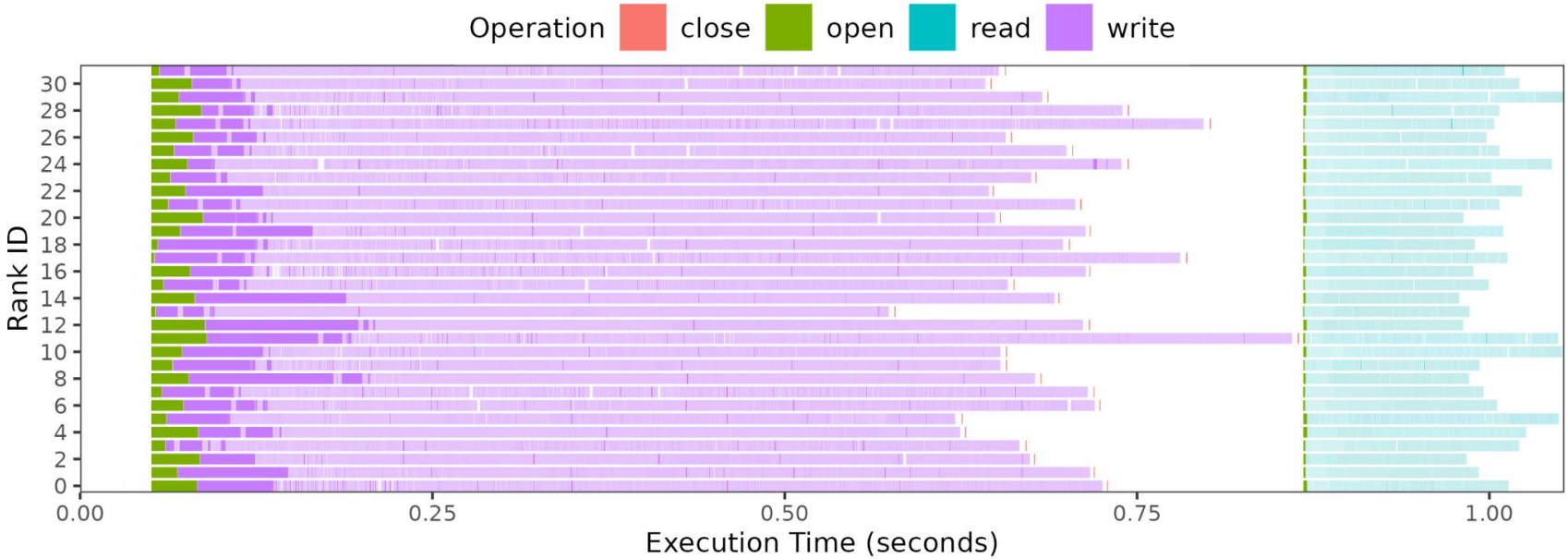
We are investigating why.

IOR - Results - LDMS-Darshan

We can identify that opens are the origin of IOR “asynchronous” writes!

Short opens before reading, longer before writing

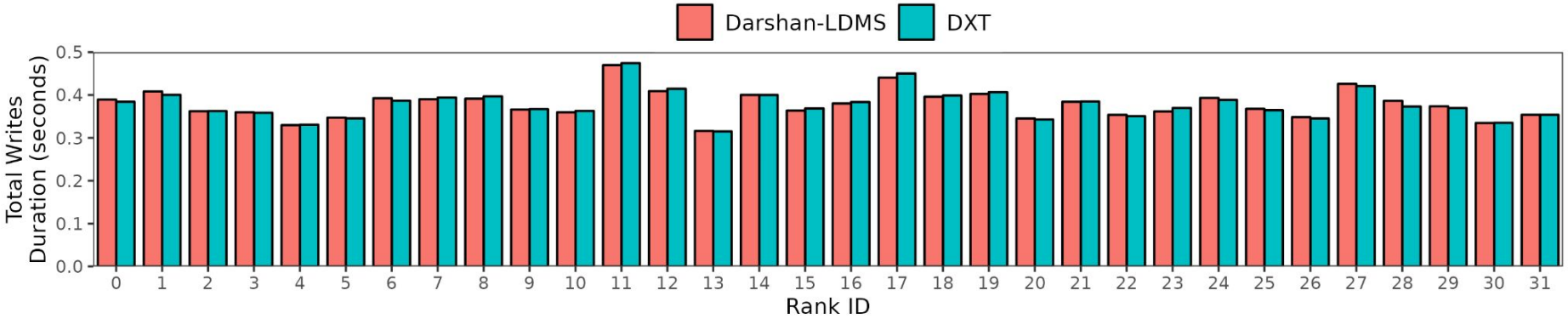
And that opens wait for all ranks writing close the file so it starts reading



IOR - Darshan-LDMS and DXT

Subtle difference between the total runtime for writes captured by Darshan-LDMS and DXT

We still can't compare reads because Darshan-LDMS is not collecting everything



Eclipse Specifications

Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz

- CPU(s): 72
- Nodes: 1,488
- L1d cache: 32K
- L1i cache: 32K
- L2 cache: 256K
- L3 cache: 46 MiB
- Total Memory: 264,047,956 kB

Total cache size per node = 2*(32*72)+256*72 + 46,080 = ~ 68.26MB
RAM: ~252GB

	total	used	free	shared	buff/cache	available
Mem(GB)	125	4	121	0	0	120