OAK RIDGE National Laboratory

Spectrum Scale (GPFS)

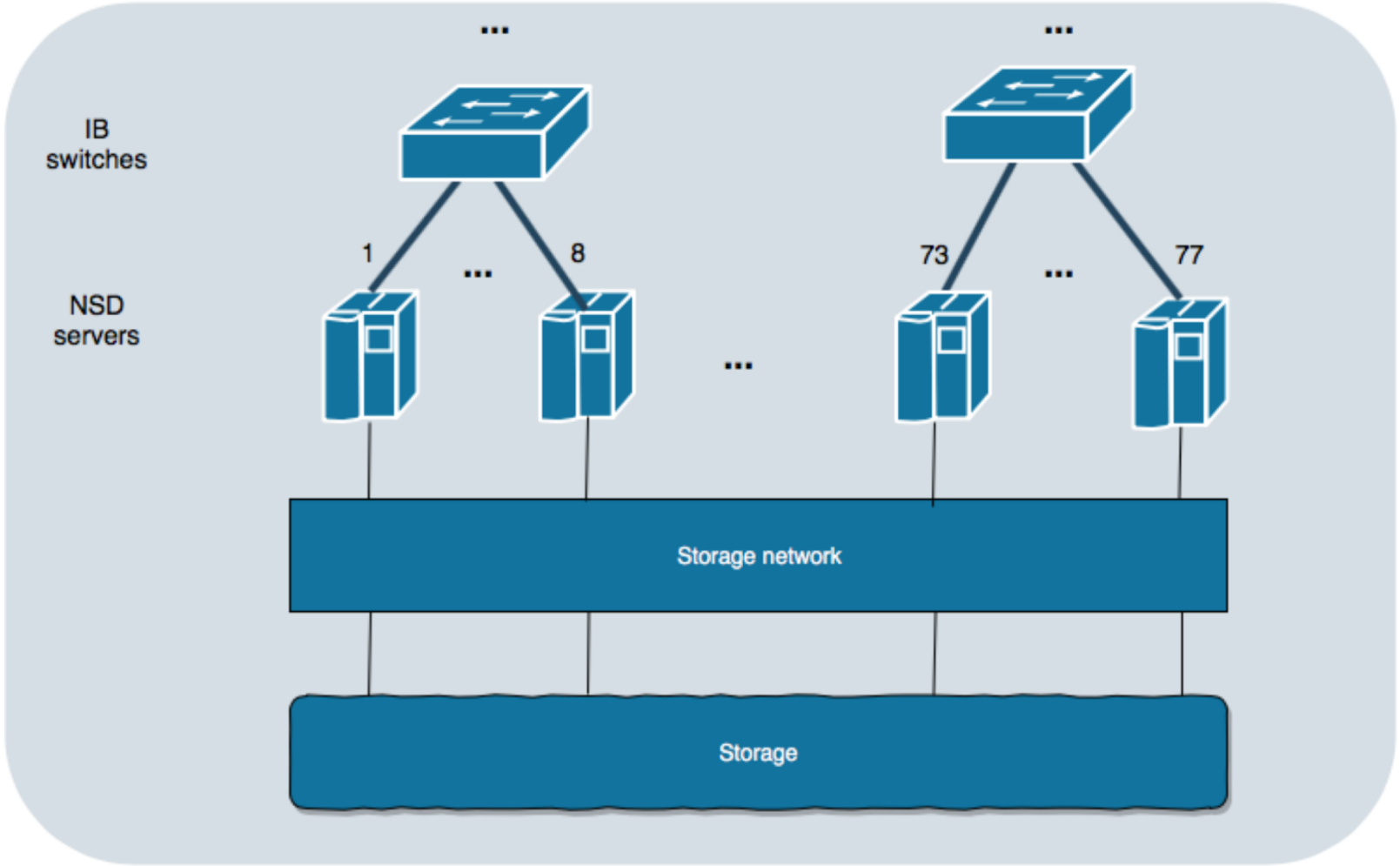
George S. Markomanolis,
HPC Engineer
Oak Ridge National Laboratory
Summit Training Workshop
Knoxville, TN
13 February 2019

U.S. DEPARTMENT OF ENERGY

Spider3 - Alpine

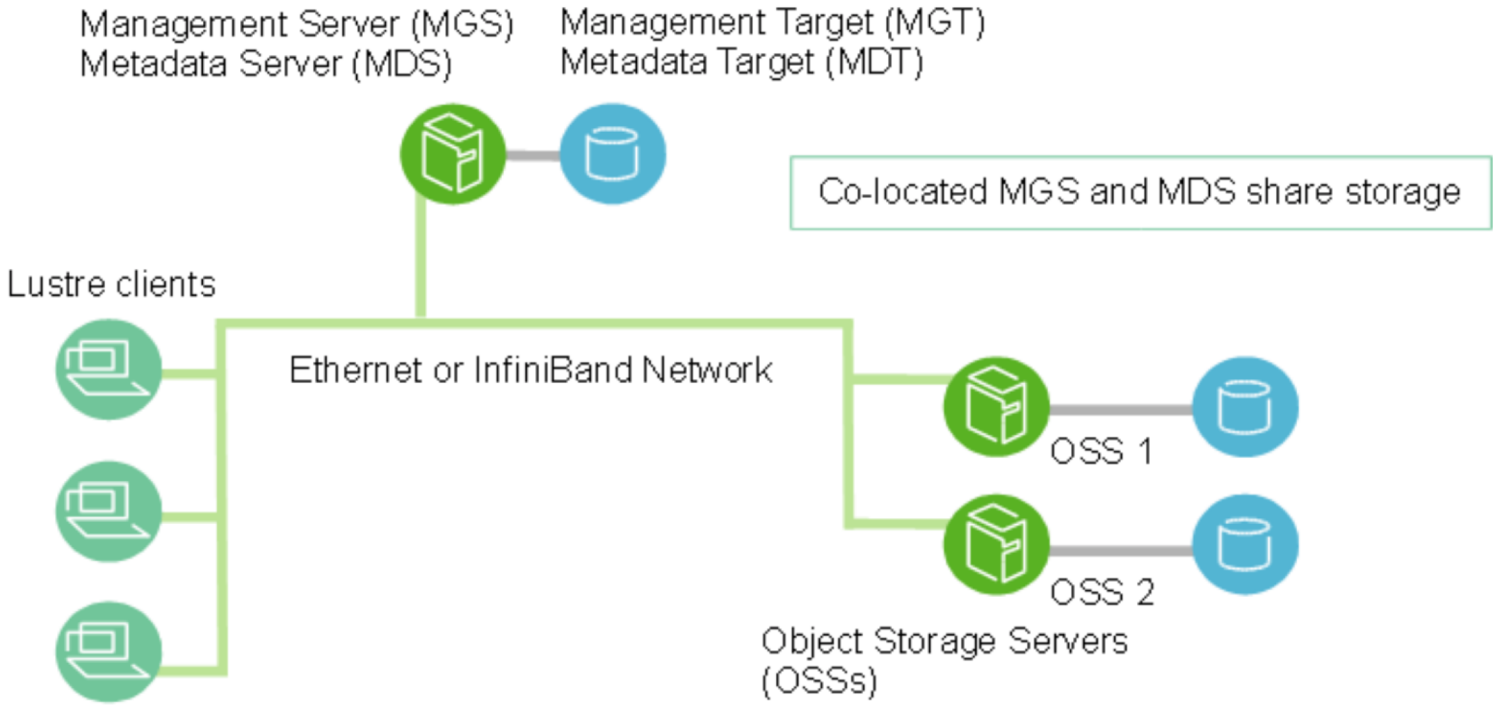
- Alpine, is a Spectrum Scale (ex-GPFS) file system of 250 PB of used space, which is mounted on Summit and Data Transfer Nodes (DTN) with maximum performance of 2.5 TB/s for sequential I/O and 2.2 TB/s for random I/O
- Largest GPFS file system installation
- Up to 2.6 million accesses per second of 32 KB small files
- It is constituted by 154 Network Shared Disk (NSD) servers
- It is a shared resource among users, supporting File Per Process (FPP), Single Shared File (SSF) and any of their combination
- EDR InfiniBand attached (100Gb/s)

Alpine – NSD servers



Atlas

- Atlas is the Lustre filesystem mounted on Titan



From Atlas to Alpine

Atlas	Alpine
User needs to stripe a folder for large files	User expects that system engineers did tune the file system
With striping, specific number of OSTs servers are used	All the NSD servers are used if the file is large enough
On Lustre there are specific number of metadata servers	On Spectrum Scale each storage server is also metadata server
On Lustre the number of the MPI I/O aggregators are equal to the number of the used OSTs	The number of the MPI I/O aggregators is dynamic, depending on the number of the used nodes

Alpine – IO-500

- IO-500 is a suite of benchmarks with 12 specific cases with purpose to extract the potential benefits of an HPC storage system based on IOR, mdtest and find tools
- During SC18, it achieved the #1 on IO-500 list, while using mainly the Spectrum Scale NLSAS and no Burst Buffer (http://io-500.org)

#	information							io500		
	institution	system	storage vendor	filesystem type	client nodes	client total procs	data	score	bw	md
									GiB/s	kIOP/s
1	Oak Ridge National Laboratory	Summit	IBM	Spectrum Scale	504	1008	zip	366.47	88.20	1522.69
2	Korea Institute of Science and Technology Information (KISTI)	NURION	DDN	IME	2048	4096	zip	160.67	554.23	46.58
3	University of Cambridge	Data Accelerator	Dell EMC	Lustre	528	4224	zip	158.71	71.40	352.75
4	JCAHPC	Oakforest-PACS	DDN	IME	2048	16384	zip	137.78	560.10	33.89

OAK RIDGE National Laboratory

What performance should we expect?

- It depends on your application and the used resources! Network could be the bottleneck if there is not enough available bandwidth available
- Results from IO-500, 504 compute nodes, 2 MPI processes per node

IOR-Write		IOR-Read	
Easy	Hard	Easy	Hard
2158 GB/s	0.57 GB/s	1788 GB/s	27.4 GB/s

- IOR Easy is I/O with friendly pattern for the storage with one file per MPI process
- IOR Hard is I/O with non-friendly pattern for the storage with a shared file
- You need always to be pro-active with the performance of your I/O

What performance should we expect? (cont.)

- It depends on the other jobs!
- There are many users on the system that they could perform heavy I/O
- The I/O performance is shared among the users

IOR Write – 10 compute nodes (not on full file system)	
Single IOR	Two concurrent IOR
144 GB/s	90 GB/s, 69 GB/s

- This is an indication that when your I/O performance does not perform as expected, you should investigate if any other large job is running with potential heavy I/O

Flags to improve I/O performance

- GPFS processes are operating only on the isolated core of each socket
- In order to give access to all the cores to handle GPFS requests, use the following option in your submission script. Usually it works better for small number of nodes

```
#BSUB -alloc_flags "smt4 maximizegpfs"
```

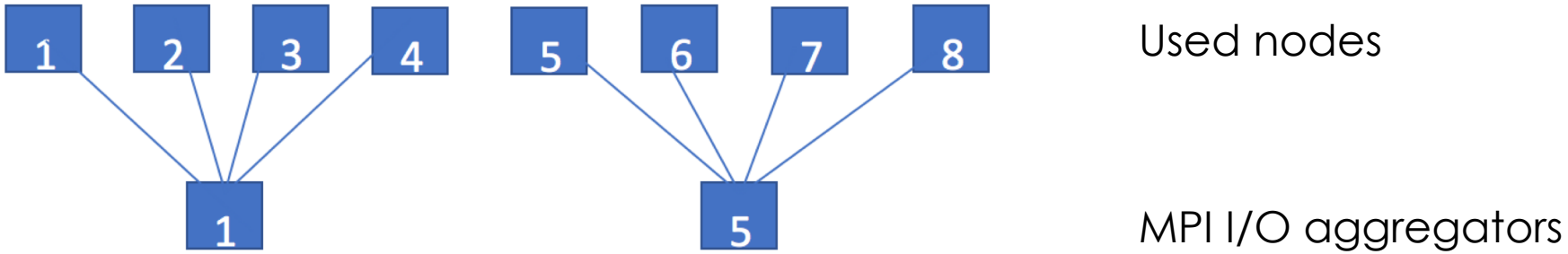
- The previous IOR write is decreased by up to 20% without the above flag
- **Important**: GPFS processes could interfere with an application, use the mentioned flag with caution and only if there is significant I/O

Spectrum Scale Internals

- Block-size: The largest size of I/O that Spectrum Scale can issue to the underlying device, on Summit it is 16MB
- All the previous IOR tests were executed with 16 MB block size
- A test with 2 MB of block-size provides write performance of 110GB/s, which is 23% less than using 16 MB of block-size.

Collective Buffering – MPI I/O aggregators

- During a collective write/read, the buffers on the aggregated nodes are buffered through MPI, then these nodes write the data to the I/O servers.
- Spectrum Scale calculates the number of MPI I/O aggregators based on the used resources. If we use 8 MPI nodes, then we have 2 MPI I/O aggregators



How to extract important information on **collective** MPI I/O

- Use the following declaration in your submission script

```
export ROMIO_PRINT_HINTS=1
```

- We have the following information in the output file for an example of 16 nodes with 16 MPI processes per node:

```
key = cb_buffer_size          value = 16777216
key = romio_cb_read           value = automatic
key = romio_cb_write          value = automatic
key = cb_nodes                value = 16
key = romio_no_indep_rw       value = false
…
key = cb_config_list          value = *:1
key = romio_aggregator_list   value = 0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240
```

NAS BTIO

- NAS Benchmarks, block Tri-diagonal solver
- Test case:
  - 16 nodes with 16 MPI processes per node
  - 819 million grid points
  - Final output file size of 156 GB
  - Version with PNetCDF support
  - Blocking collective MPI I/O, single shared file among all the processes
- Write speed: **1532** MB/s
- That's significant low performance although the I/O pattern is not friendly for most of the filesystems

NAS BTIO – Block size

- The default block size for Parallel NetCDF is 512 bytes when the striping_unit is not declared
- We create a file called romio_hints with the content:

```
striping_unit 16777216
```

- Then we define the environment variable ROMIO_HINTS pointing to the file romio_hints

```
export ROMIO_HINTS=/path/romio_hints
```

- New I/O write performance is **13602** MB/s
- Speedup of **8.9!!** times for the specific benchmark without editing or compiling the code

NAS BTIO – Non-blocking PNetCDF

- We test the version with **non-blocking** collective PNetCDF with 16 nodes and 16 MPI processes per node.
- Default parameters provide write performance of **13985** MB/s, almost as the optimized blocking version.
- The exact same optimizations, provide **28203** MB/s, almost double performance.
- As the parallel I/O is non blocking we can increase the MPI I/O aggregators to evaluate the performance and we concluded that adding in the romio_hints the following command improves the performance

```
cb_config_list *:8
```

- With the above declaration, we have 8 MPI I/O aggregators per node and the performance now is **35509** MB/s, **2.54** times improved compare to the default results.

Darshan on Summit

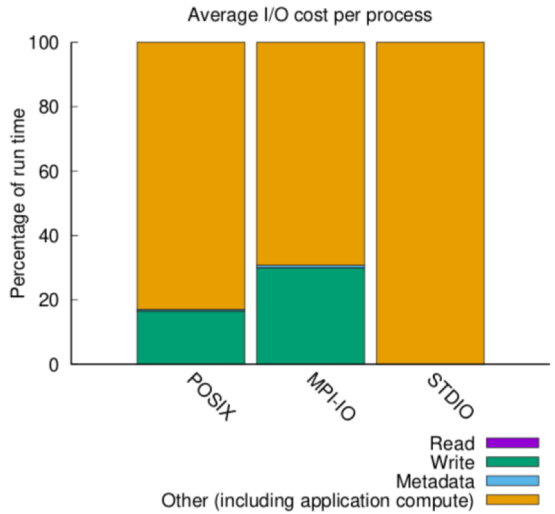
- Darshan is a lightweight I/O profiling tool
- It profiles parallel I/O, there are some developments for
- It is loaded by default on Summit since early-mid January
- If you are using Summit, probably you have already Darshan files if you are using MPI.
- Log files location: /gpfs/alpine/darshan/summit/year/month/day/${username}_binary_name_...
- If you are using HDF5 and you would like Darshan to profile the corresponding calls
  - For HDF5 v1.8: *module load darshan-runtime/3.1.6-hdf5pre110*
  - For HDF5 v1.10.x: *module load darshan-runtime/3.1.6-hdf5post110*

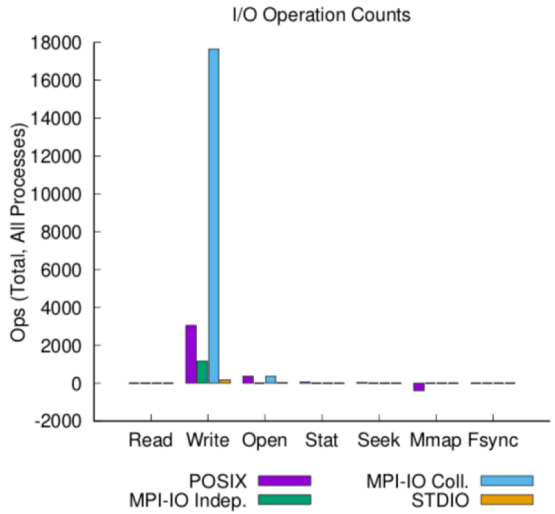
- **Important**: You have to unload the module darshan-runtime if you are using other profiling tools, even the MPI profiling from Spectrum MPI, probably the application will crash or stall

Darshan on Summit

jobid: 244162	uid: 14850	nprocs: 36	runtime: 21 seconds

I/O performance *estimate* (at the MPI-IO layer): transferred 8770 MiB at 1519.52 MiB/s
I/O performance *estimate* (at the STDIO layer): transferred 0.0 MiB at 65.28 MiB/s





Darshan on Summit (cont.)

Most Common Access Sizes (POSIX or MPI-IO)

	access size	count
POSIX	16777216	1250
	272	200
	9437184	90
	4608	90
MPI-IO ‡	262144	6840
	128	3240
	12800	2880
	1280	2520

‡ NOTE: MPI-IO accesses are given in terms of aggregate datatype size.

File Count Summary (estimated by POSIX I/O access offsets)

type	number of files	avg. size	max size
total opened	11	2.0G	2.2G
read-only files	0	0	0
write-only files	11	2.0G	2.2G
read/write files	0	0	0
created files	11	2.0G	2.2G

Darshan on Summit (cont.)

Variance in Shared Files (POSIX and STDIO)

File Suffix	Processes	Fastest			Slowest			σ	
		Rank	Time	Bytes	Rank	Time	Bytes	Time	Bytes
...000000002.h5	36	3	0.000281	872	0	1.472090	497M	0.296	1.49e+08
...000000006.h5	36	9	0.010528	856	0	1.414556	497M	0.252	1.49e+08
...000000003.h5	36	3	0.000258	872	0	1.353647	497M	0.268	1.49e+08
...000000009.h5	36	1	0.000271	704	0	1.351178	497M	0.288	1.49e+08
...000000004.h5	36	1	0.000273	704	0	1.322674	497M	0.279	1.49e+08
...000000010.h5	36	5	0.000270	520	0	1.321882	497M	0.296	1.49e+08
...000000007.h5	36	2	0.000231	424	0	1.315145	497M	0.299	1.49e+08
...000000005.h5	36	3	0.000301	872	0	1.235631	497M	0.284	1.49e+08
...000000001.h5	36	5	0.000306	520	0	1.224918	497M	0.269	1.49e+08
...000000008.h5	36	3	0.000332	872	0	1.114605	497M	0.298	1.49e+08
...<STDOUT>	36	1	0.000000	0	0	0.000128	8.6K	0	1.44e+03

Conclusion

- Use parallel I/O libraries that are optimized such as ADIOS, PNetCDF, HDF5 etc.
- Use non-blocking MPI I/O to improve the performance
- Do not re-invent the wheel!
- Remember that Alpine is a shared resource
- Use tools that provide insight I/O performance information such as Darshan

Acknowledgement

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

Thank you!

Questions?