OAK RIDGE National Laboratory

Summit storage

George S. Markomanolis,

HPC Engineer

Oak Ridge National Laboratory

OLCF User Conference Call

12 December 2018

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Outline

- Storage Areas/ Data Transfer
- Introduction to Spectrum Scale
- Introduction to Burst Buffer
- Burst Buffer libraries

Open slide master to edit

OAK RIDGE National Laboratory

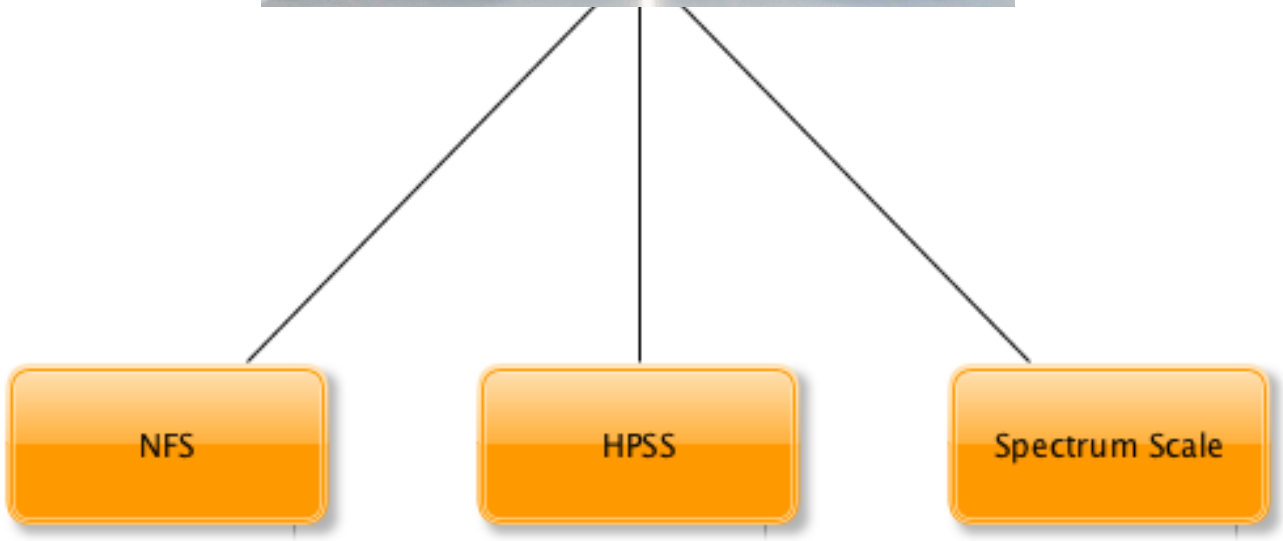
Storage Areas/ Data Transfer

Outline

- Storage Areas
  - Available file systems and options for archiving
- Data Transfer
  - Transfer your files between Titan and Summit

Open slide master to edit

Summit and filesystems



NFS

- User home: /ccs/home/$USER
- Project home: /ccs/proj/[projid]
- **Long-term** storage for your general data under home or related to project under proj
- **Build** your code in /tmp/$USER it is faster and **install** in /ccs/proj/[projid]
- There is provided a **backup**
- User home and project home are accessible read-only from the Summit compute nodes
- **Not purged**
- **Quota** of 50GB
- User home is user-centric

Open slide master to edit

NFS (cont.)

- Check quota on user home

```
> quota -Qs
Disk quotas for user gmarkoma (uid 14850):
Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
nccs-svm1.lb.ccs.ornl.gov:/nccs/home2
                3237M  51200M  51200M          49161   4295m   4295m
```


OAK RIDGE National Laboratory
Open slide master to edit

NFS (cont.)

- I deleted a file from my NFS, how to recover it?
- Answer: snapshots
  - Go to the .snapshot folder (ls will not show this folder):
  - cd .snapshot

```
ls -l
drwx------ 27 gmarkoma gmarkoma 4096 Nov 21 16:51 daily.2018-11-23_0010
drwx------ 27 gmarkoma gmarkoma 4096 Nov 21 16:51 daily.2018-11-24_0010
…
```


OAK RIDGE National Laboratory
Open slide master to edit

HPSS

- User archive: /home/$USER
- Project archive: /proj/[projid]
- **Long-term** storage for large amount of general data under home or related to project under proj.
- **Quota** of 2 TB and 100 TB for user and project archive respectively. If any of the used files during htar is bigger than 68 GB size, then it will fail, similar if there are more than 1 million files per archive
- **Not purged**
- User archive is user-centric

Open slide master to edit

HPSS (cont.)

- Check HPSS quota (this moment from DTN or Titan):

```
> showusage -s hpss
HPSS Storage in GB:
                    Project Totals
 Project                 Storage          Storage
_____________________|_____________________|_____________
 stf007              |    46868.90         |       0.00
```

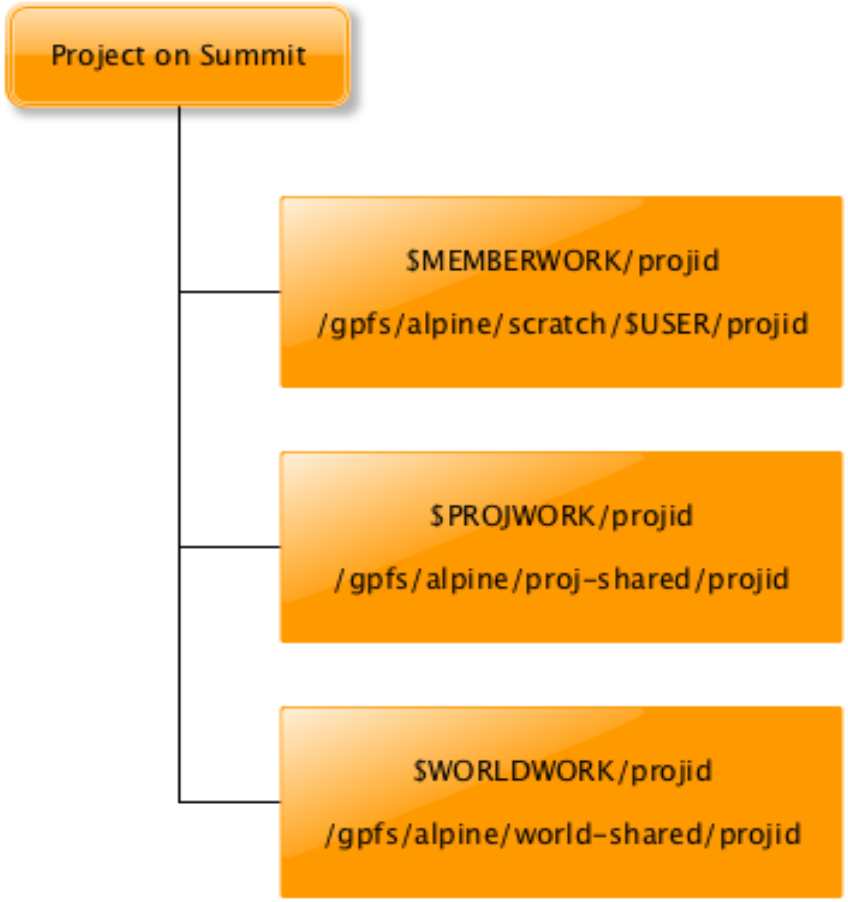

OAK RIDGE National Laboratory
Open slide master to edit

Spider III - 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)

Open slide master to edit

Alpine (cont.)



- Memberwork:
  - Short-term storage of user data related to the project but not shared
- Projwork:
  - Short-term storage of project data shared among the members of the project
- Worldwork:
  - Short-term storage of project data shared with OLCF users outside the project
- **No backup**
- **Quota 50 TB**
- **Purged after 150 days**

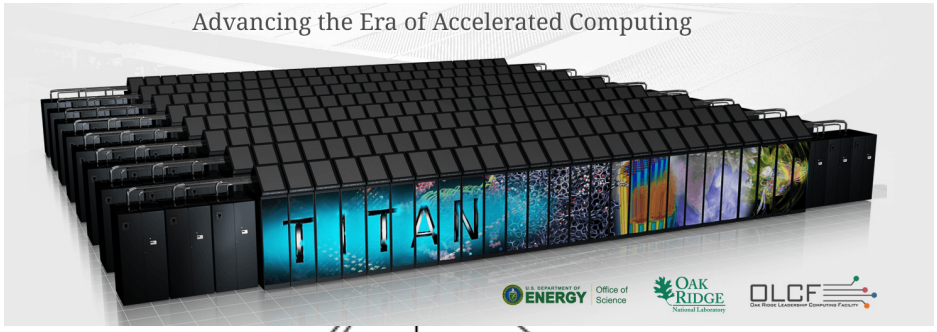
Open slide master to edit

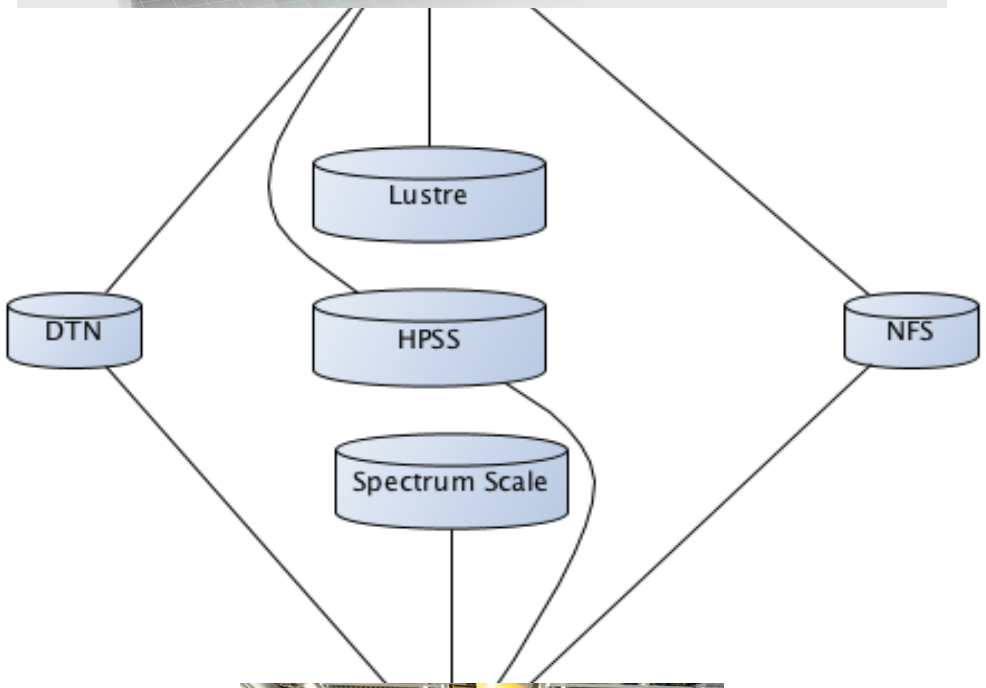
Storage policy

Name	Path	Type	Permissions	Backups	Purged	Quota
User Home	`$HOME`	NFS	User Set	yes	no	50GB
User Archive	`/home/$USER`	HPSS	User Set	no	no	2TB
Project Home	`/ccs/proj/[projid]`	NFS	770	yes	no	50GB
Member Work	`/gpfs/alpine/scratch/[userid]/[projid]/`	Spectrum Scale	700	no	150 days	50TB
Project Work	`/gpfs/alpine/proj-shared/[projid]`	Spectrum Scale	770	no	150 days	50TB
World Work	`/gpfs/alpine/world-shared/[projid]`	Spectrum Scale	775	no	150 days	50TB
Project Archive	`/proj/[projid]`	HPSS	770	no	no	100TB

Open slide master to edit

Data Transfer





Data Transfer Nodes (DTN) improve the performance by reducing the load on the login and service nodes of the HPC facilities. Moreover, transfer data outside the HPC facility.

Data Transfer (cont.)

- When you log-in to Summit you would like to have access to your old files (if you are already user of OLCF HPC facilities)
- There are many ways to transfer files but in general we propose Globus
- We will mention all the approaches and some performance results.

Advices about transferring files

- Start as soon as possible, many users probably will transfer files on the same moment
- Titan and Atlas will be available up to the end of September 2019
- It's time to clean your data!
- The data that you are not going to use soon, but you need, save them to HPSS and delete them from Atlas.

Open slide master to edit

Data Transfer - NFS

- If the data size is less than 50 GB and there is enough free space in your home directory

```
titan> cp –r data $HOME

summit> cp –r $HOME/data .
```

- It is simple, but is it fast?

OAK RIDGE National Laboratory
Open slide master to edit

Data Transfer - HPSS

- Using HPSS
- Send one folder to HPSS and retrieve it from the destination. There is significant higher data size limit

titan> htar -cvf transfer_test.tar transfer_test/*

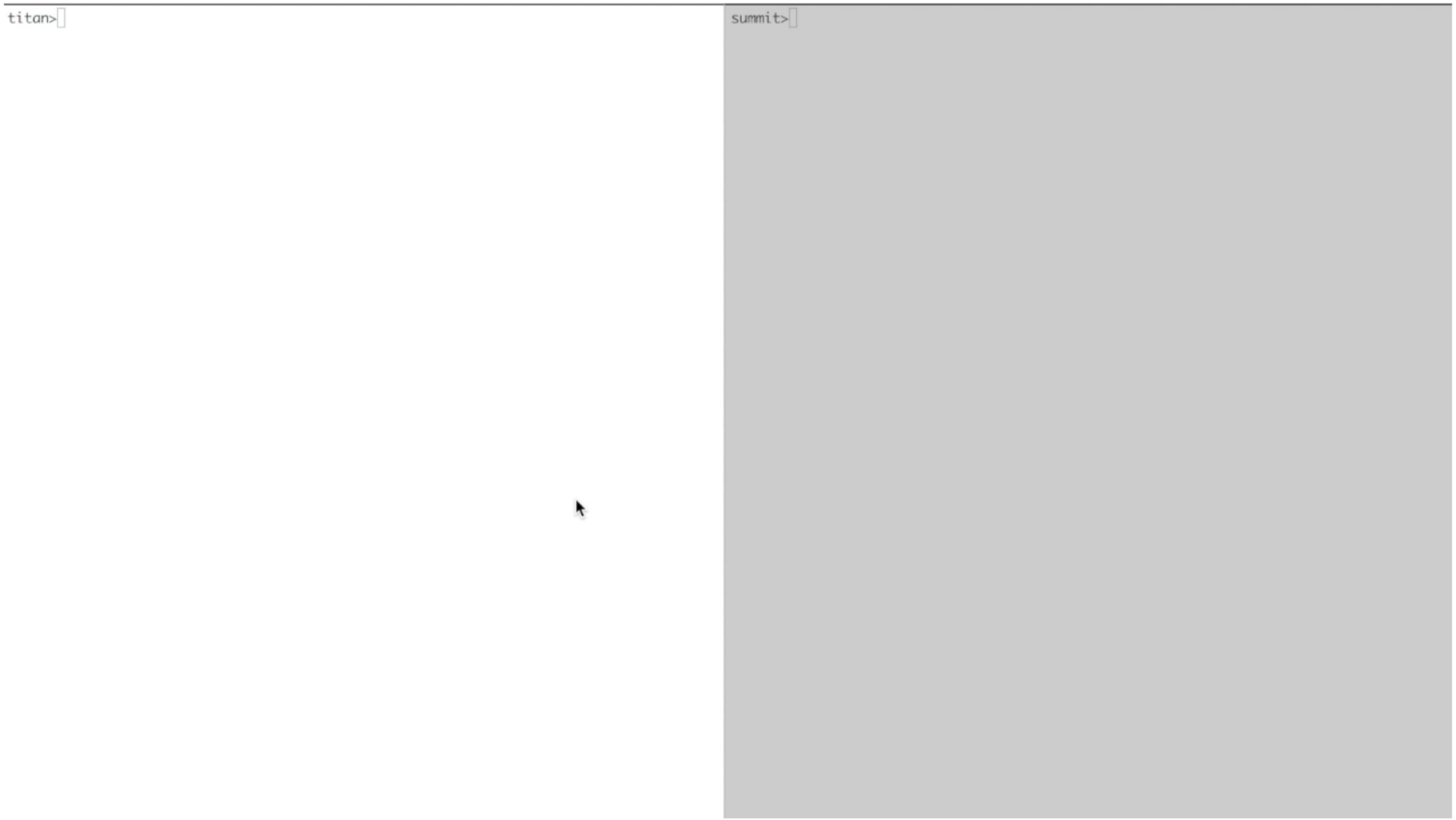
HTAR: a transfer_test/data0.txt
HTAR: a transfer_test/data10.txt
…
HTAR: a /tmp/HTAR_CF_CHK_8183_1543522594
HTAR Create complete for transfer_test.tar. 23,068,684,800 bytes written for 22 member files, max threads: 3 Transfer time: 186.324 seconds (123.809 MB/s) wallclock/user/sys: 186.521 30.654 105.275 seconds

HTAR: HTAR SUCCESSFUL

summit> htar -xvf transfer_test.tar

OAK RIDGE National Laboratory
Open slide master to edit

Transferring files through NFS and HPSS



Open slide master to edit

Globus

- Globus transfers fast, parallel and reliable files between two endpoints
- Endpoints are different locations where data can be moved using the Globus transfer
- Visit www.globus.org to login
- You can find few OLCF endpoints such as **OLCF DTN**. Since 11th December the endpoint OLCF Atlas is renamed to **OLCF DTN**.

Open slide master to edit

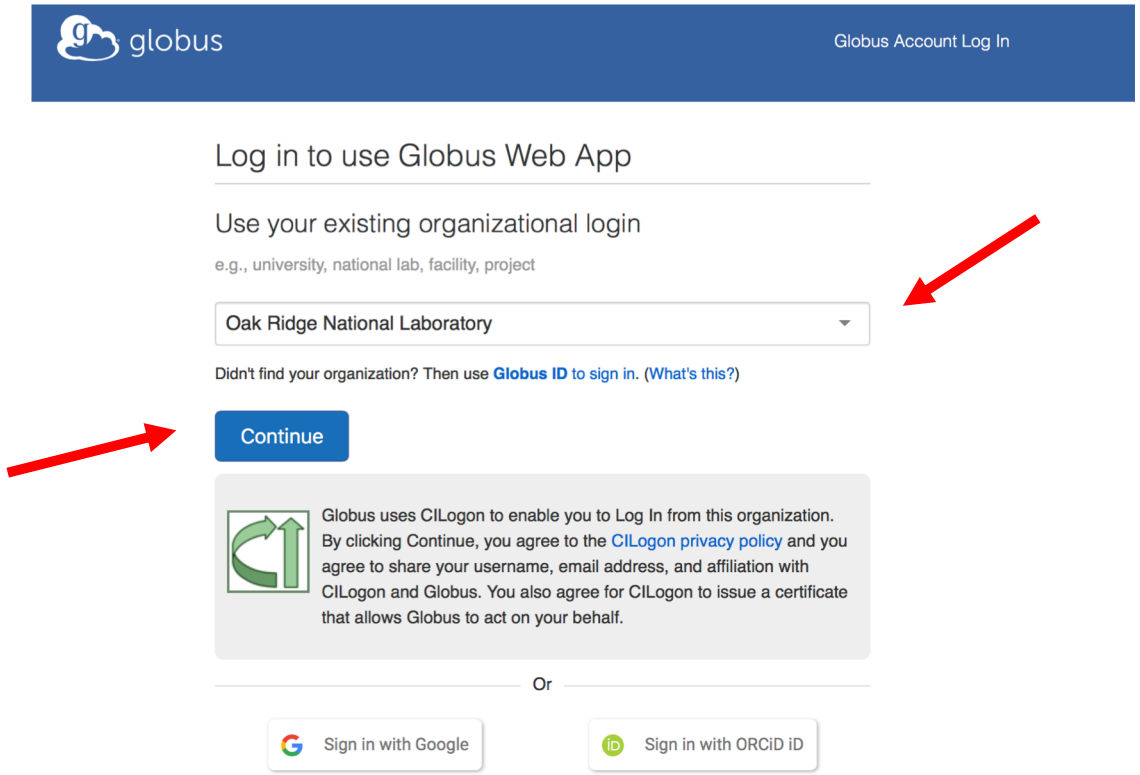
Globus demo, transfer from Titan to Summit

Open slide master to edit

Globus (www.globus.org)

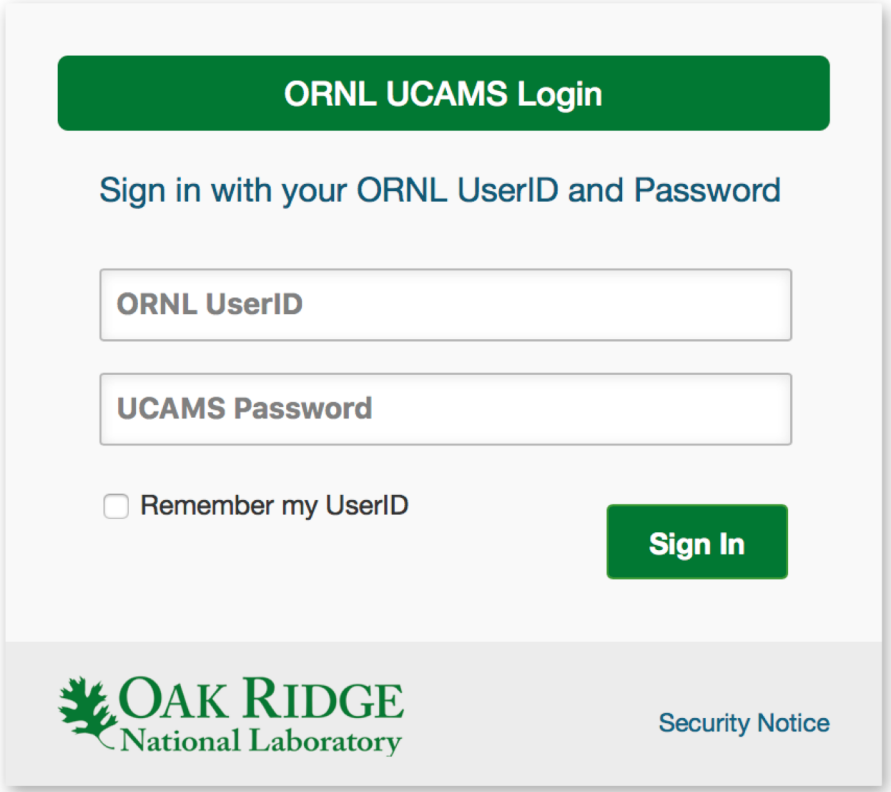
Open slide master to edit

Select your organization

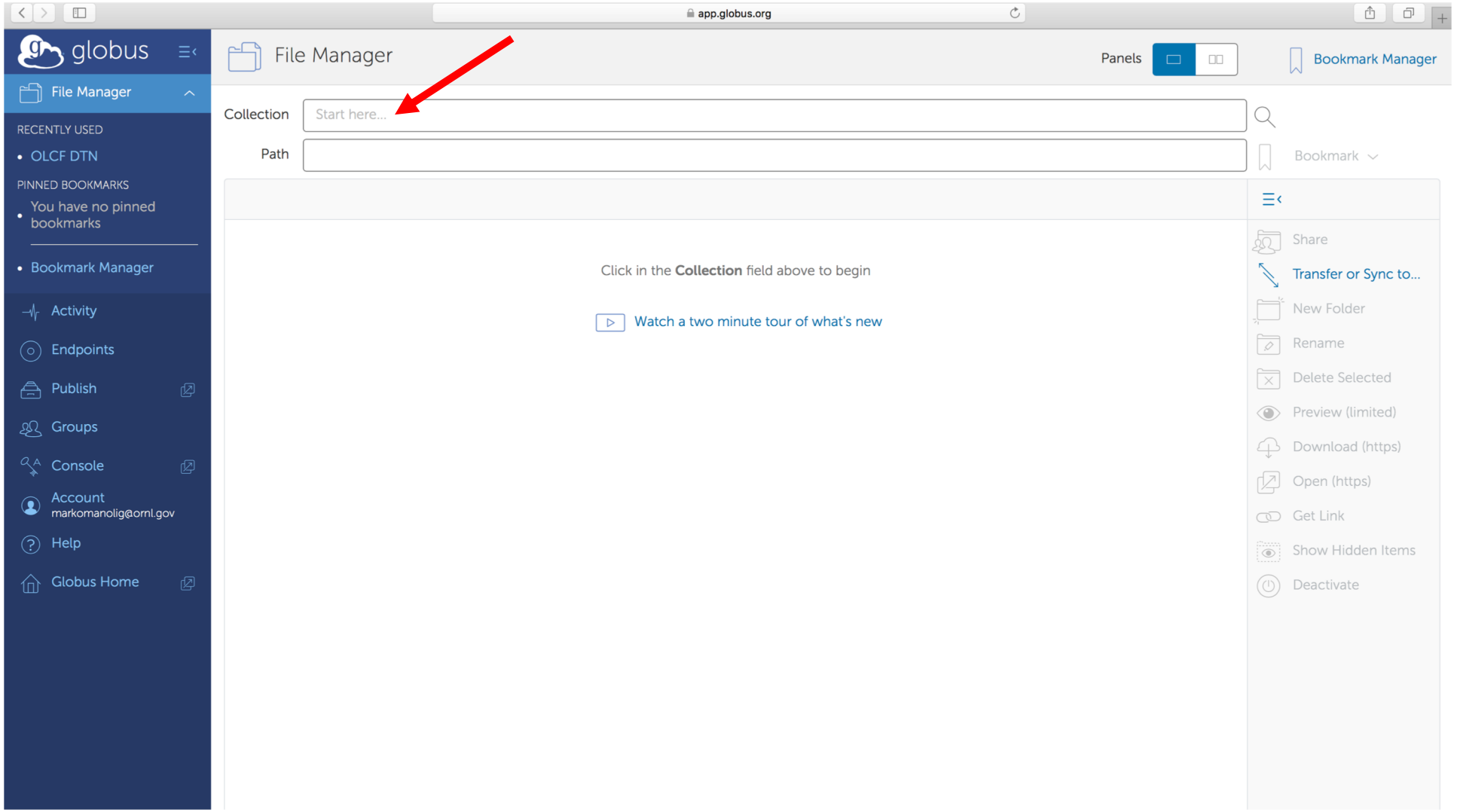


Open slide master to edit

Credentials



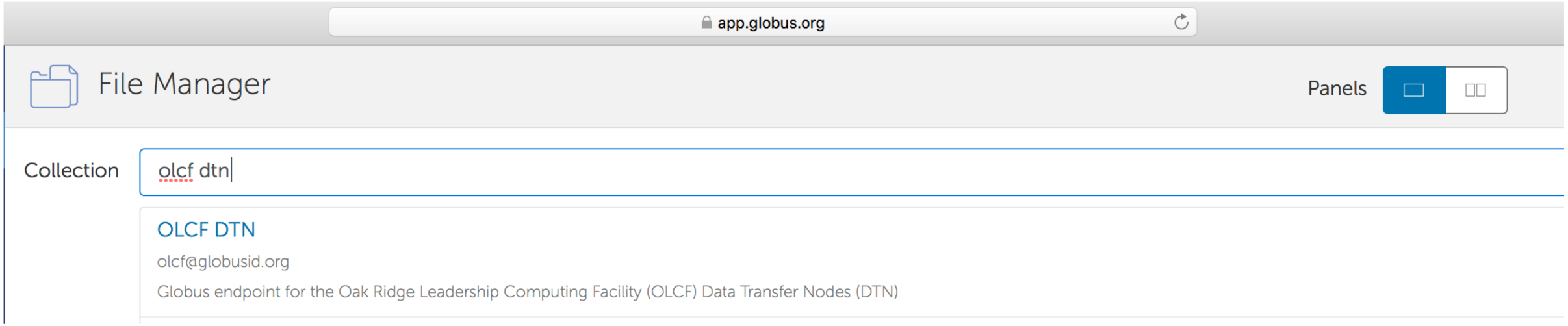
Select an endpoint



OAK RIDGE National Laboratory

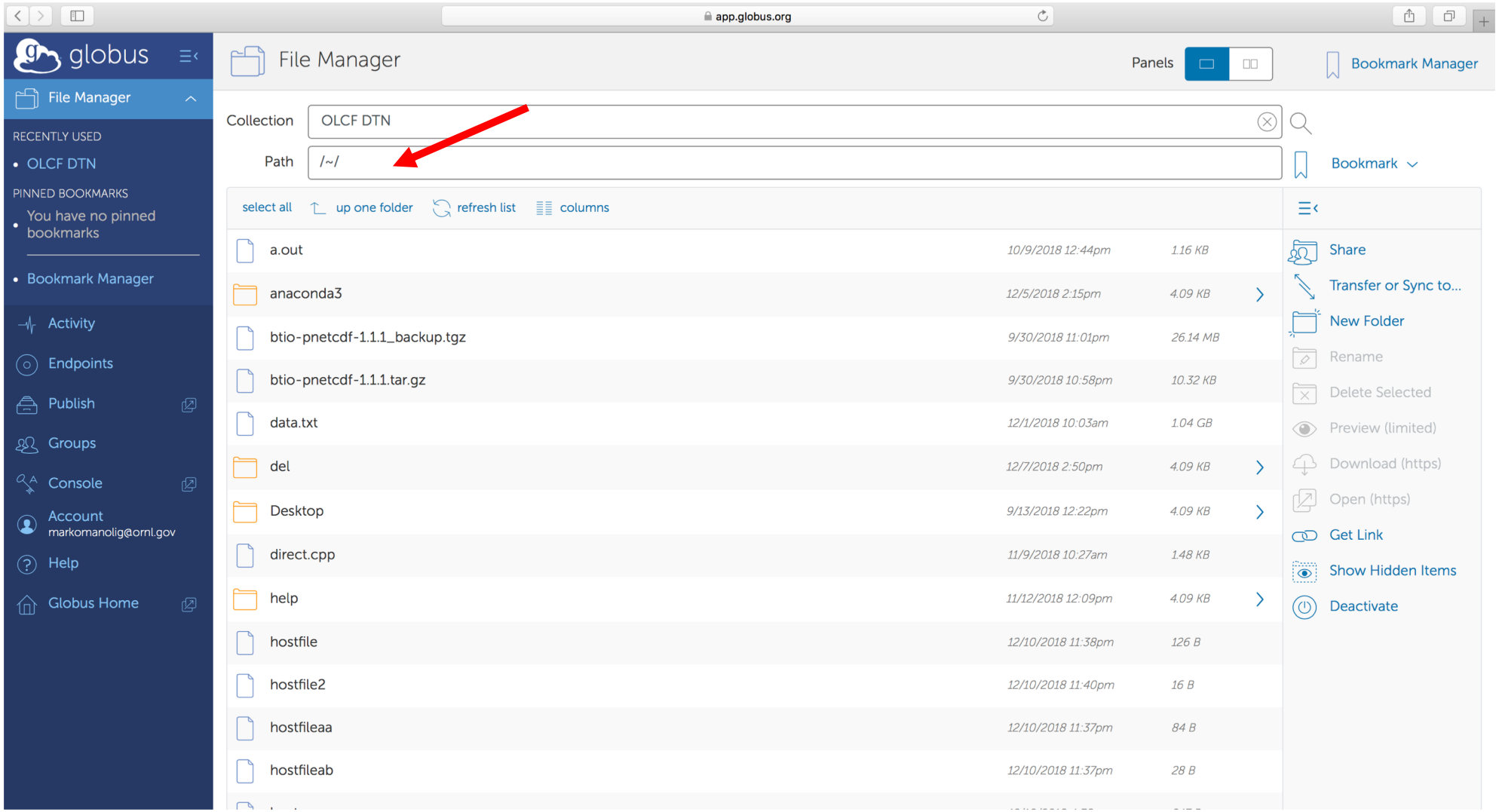
Open slide master to edit

Search for an endpoint



Open slide master to edit

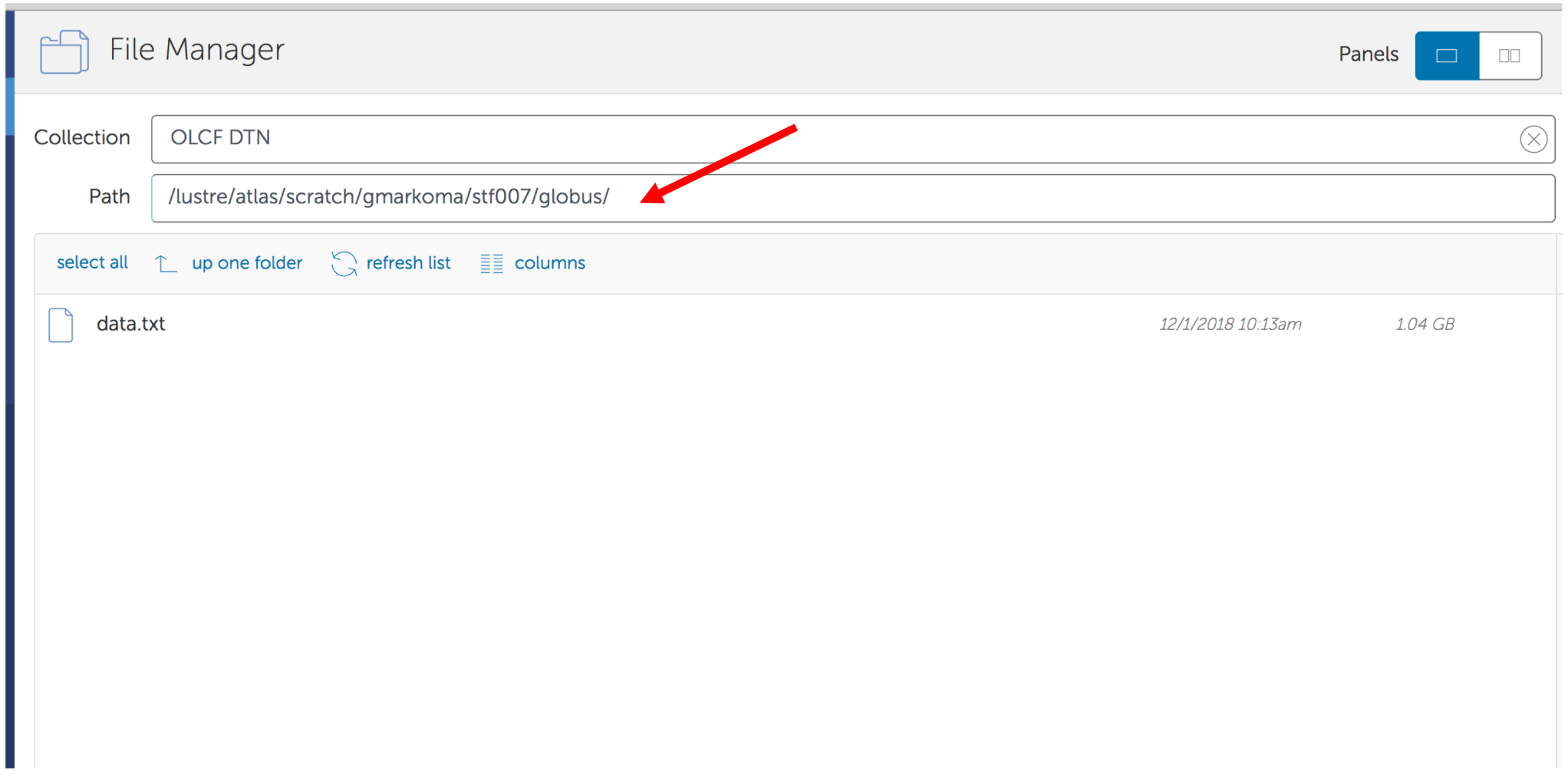
Find the path with the required files



OAK RIDGE National Laboratory

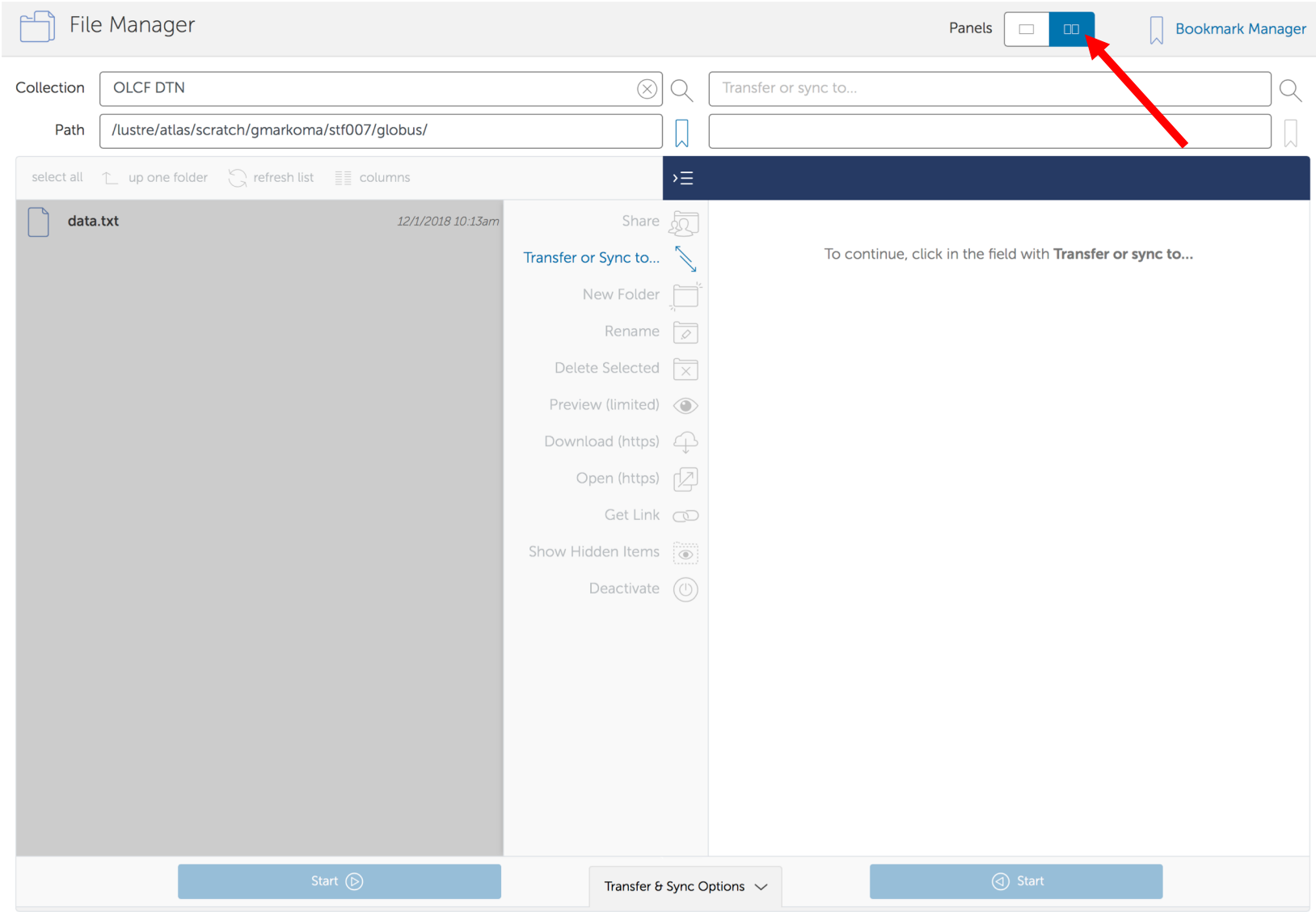
Open slide master to edit

Find the path with the required files (cont.)



Open slide master to edit

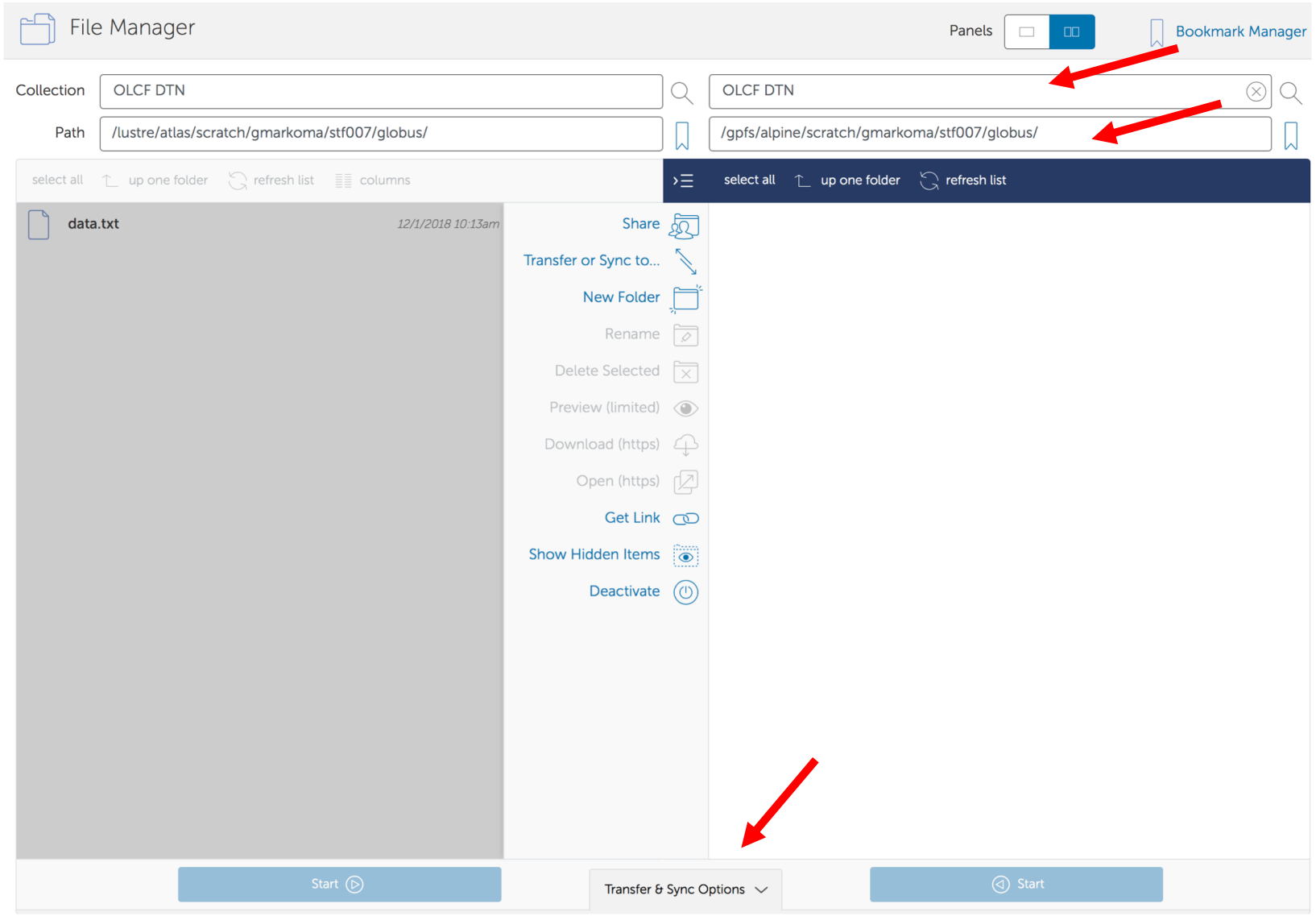
Choose appropriate panels option



OAK RIDGE National Laboratory

Open slide master to edit

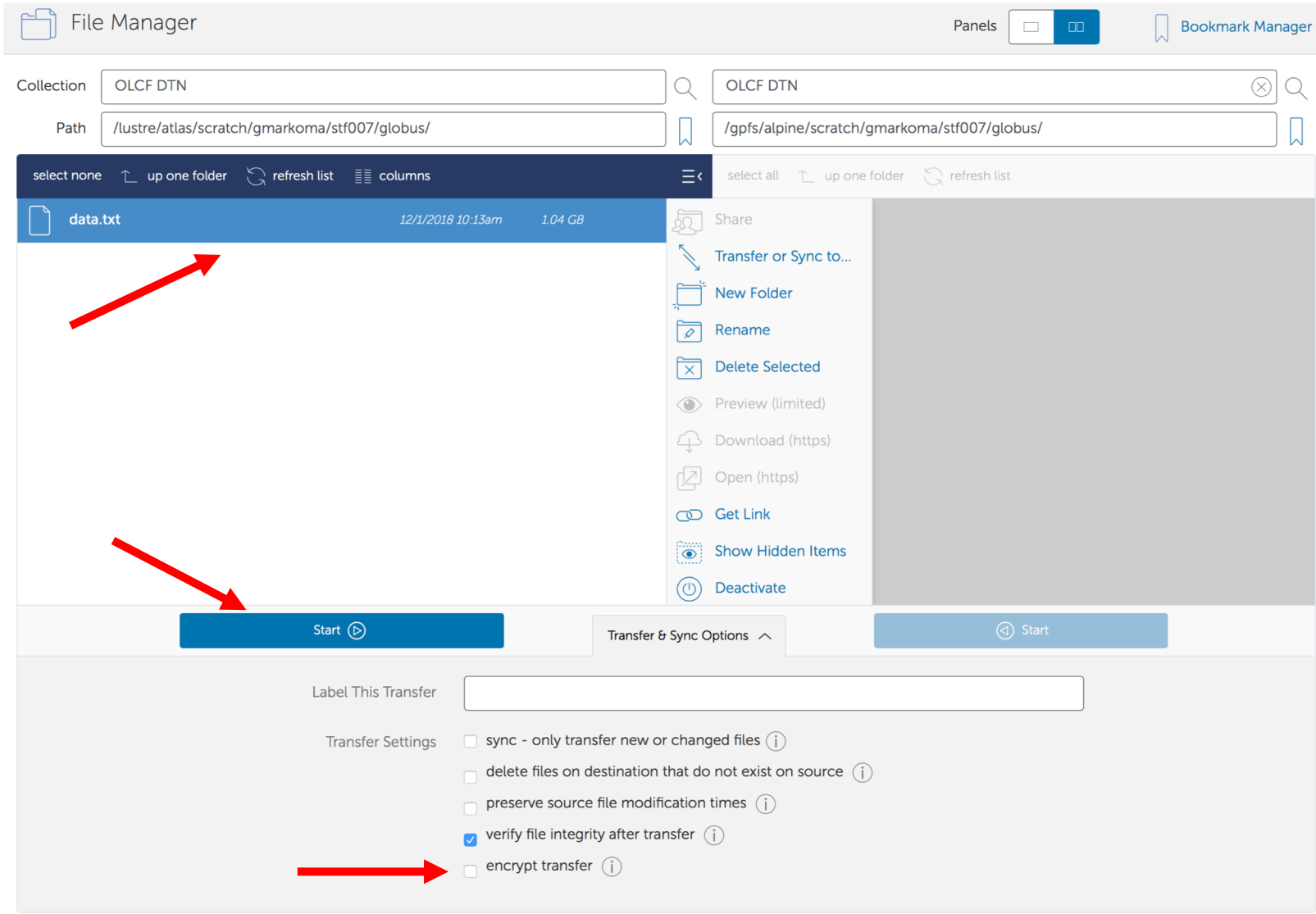
Use appropriate settings



OAK RIDGE National Laboratory

Open slide master to edit

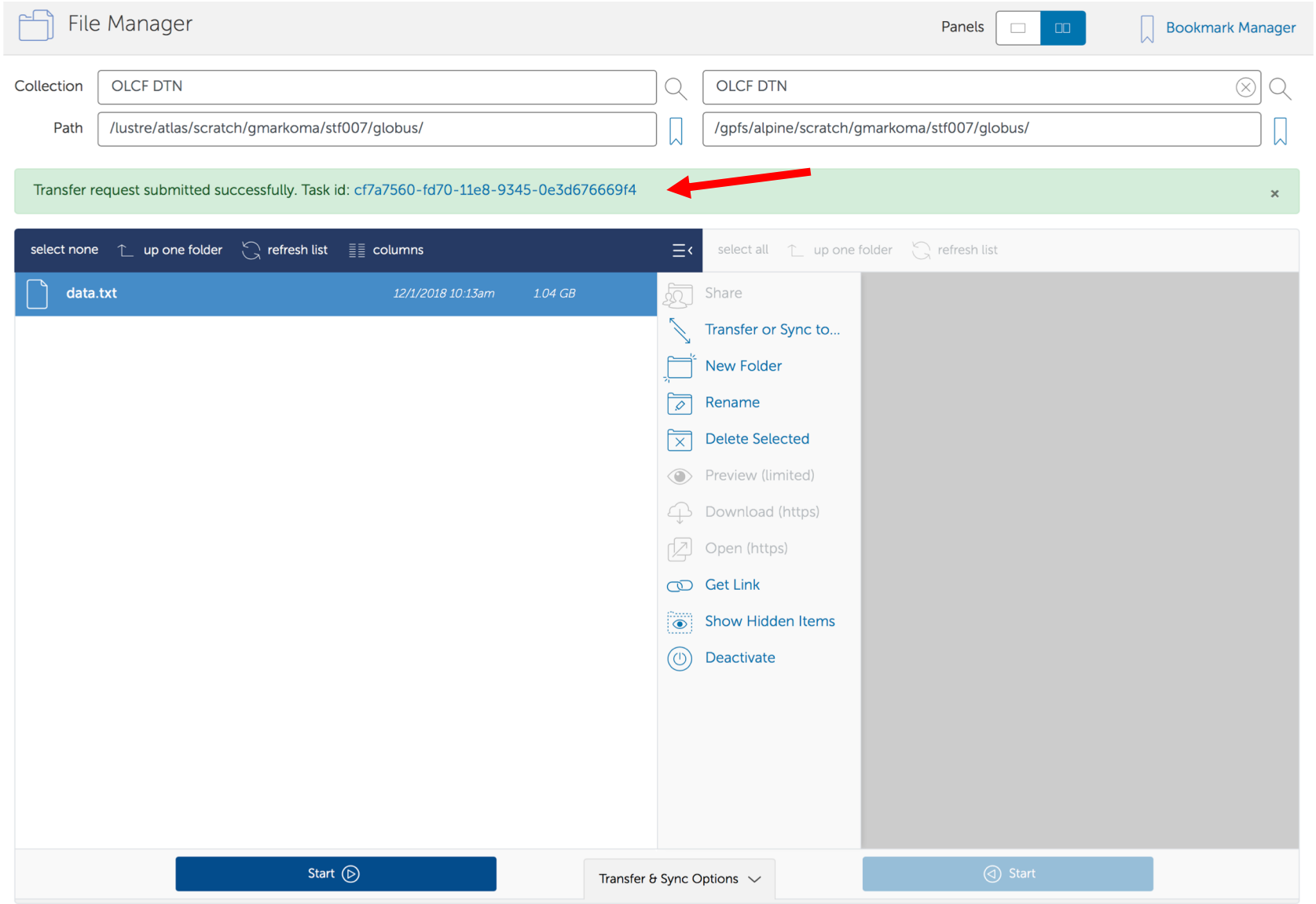
Transfer encryption



OAK RIDGE National Laboratory

Open slide master to edit

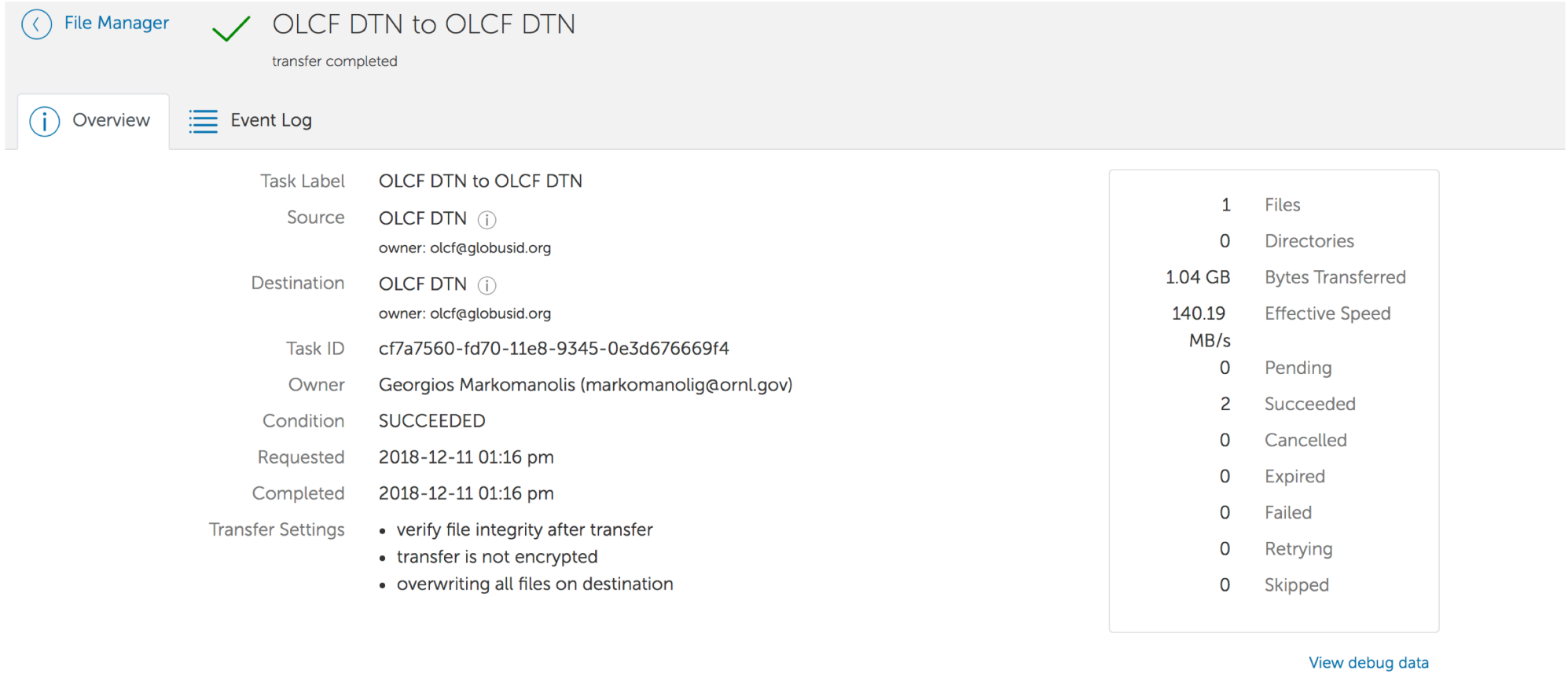
Activity



OAK RIDGE National Laboratory

Open slide master to edit

Activity report



Open slide master to edit

Performance Results

- Study case: Transfer data from Atlas to Alpine with 3 approaches. Copy the files through NFS, use HPSS, or use Globus

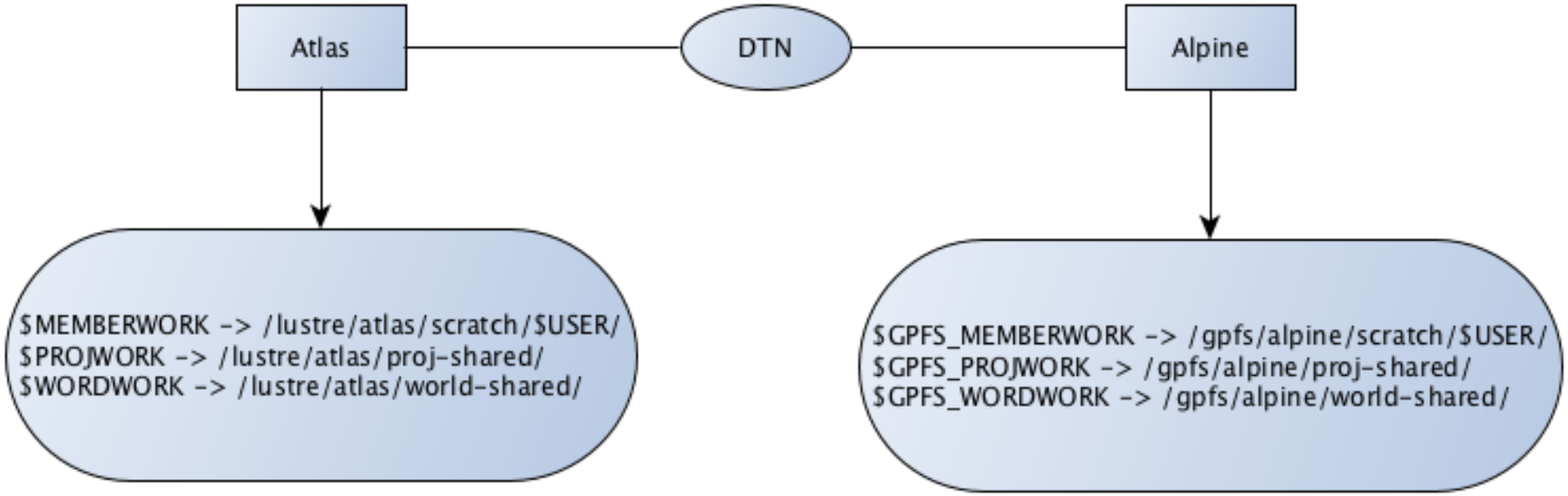
Type	Home NFS	HPSS	Globus
	Time in seconds to finish the transfer		
Transfer 22 files of 1GB each	323	270	10
Transfer 1 file of 22 GB	308	301	80
Transfer 4 files of 1GB each	69	53	9

- Globus is the most efficient approach to transfer files for all the evaluated cases, for small files though, transferring through NFS should be efficient.
- There are available some traditional tools such as scp, rsync
- The tests took place on 29th November

Open slide master to edit

DTN

- As long as we have both Atlas and Alpine on DTN, we use the following variables (Alpine is not mounted on all DTN nodes yet)



Open slide master to edit

Conclusions – Storage areas/Data transfer

- Use NFS for installing your libraries (long-term storage)
- There are many approaches to transfer files, it seems that Globus is the fastest one but it depends on the number of files, file size etc.
- Use HPSS for large files that you don't plan to use soon and to backup soon to expire projects with important data
- Start transferring your files to Summit as soon as you have access
- Do not forget the storage policy!

Open slide master to edit

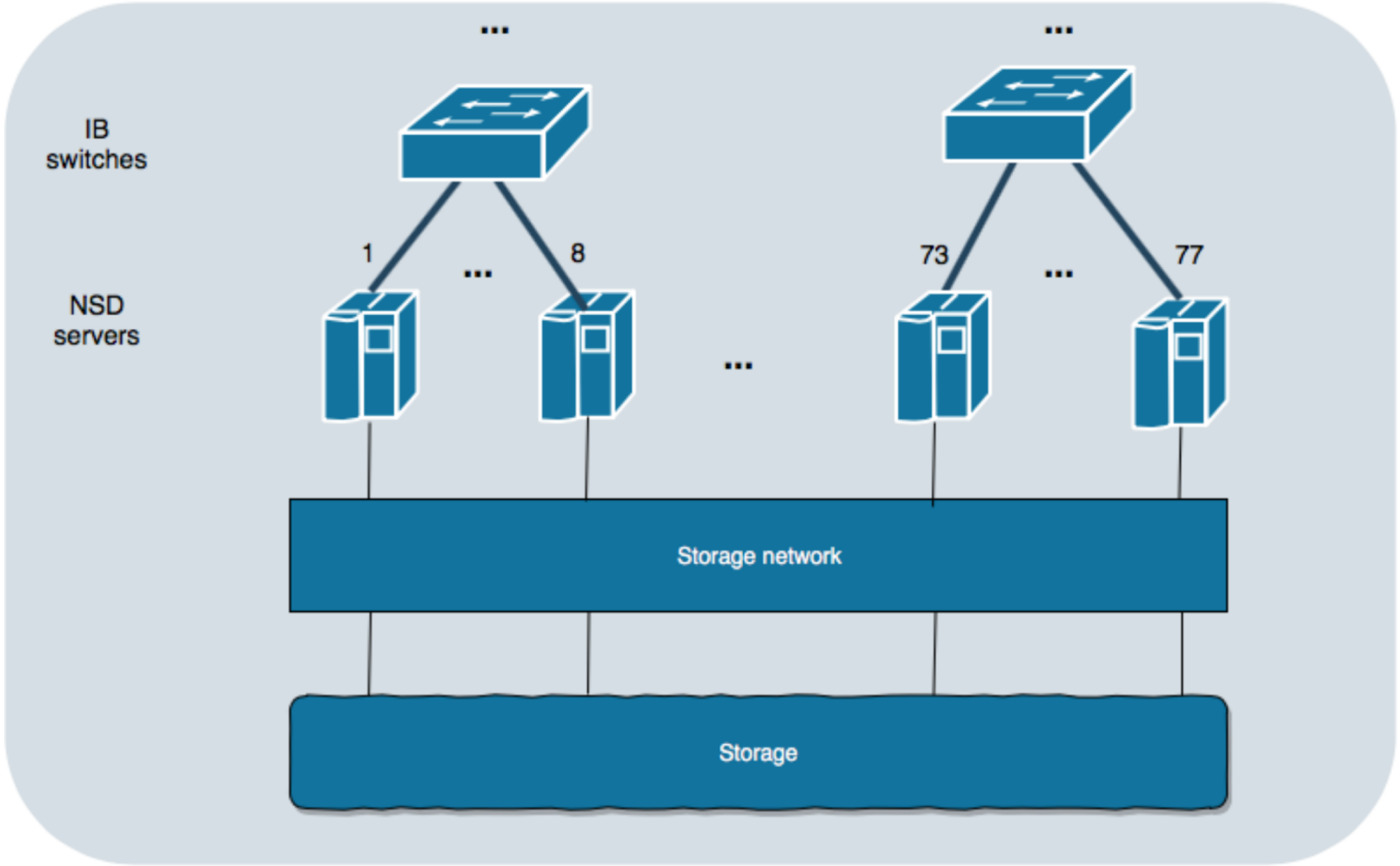
OAK RIDGE National Laboratory

Spectrum Scale

Spider III - 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)

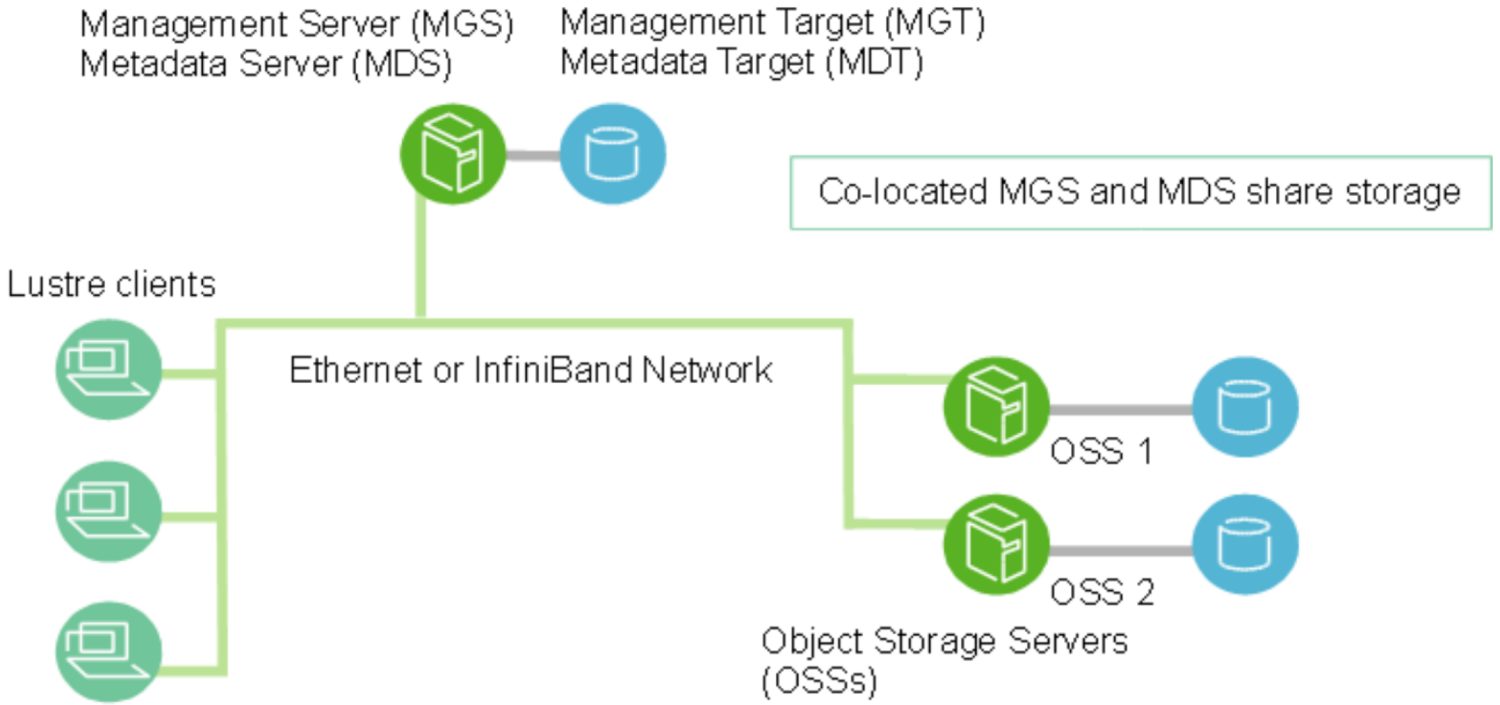
Alpine – NSD servers



Open slide master to edit

Atlas

- Atlas is the Lustre filesystem mounted on Titan



Open slide master to edit

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

Open slide master to edit

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

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

Open slide master to edit

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

Open slide master to edit

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

```
#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

Open slide master to edit

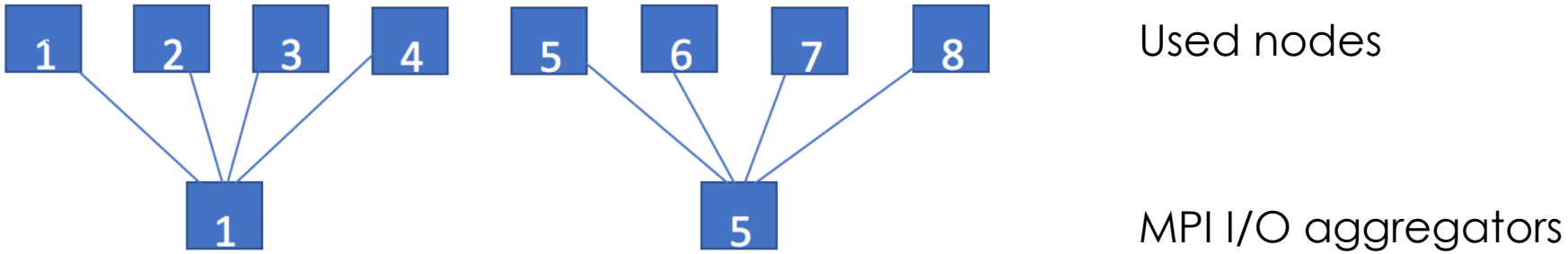
Spectrum Scale Internals

- 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.

Open slide master to edit

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 compute nodes, then we have 2 MPI I/O aggregators



OAK RIDGE National Laboratory
Open slide master to edit

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
```


OAK RIDGE National Laboratory
Open slide master to edit

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

Open slide master to edit

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

OAK RIDGE National Laboratory
Open slide master to edit

NAS BTIO – Hints

- Update the file romio_hints and define

```
romio_no_indep_rw true
```

- Then the processes that are not MPI I/O aggregators, they will not open the output file as they are not going to save any data on it

- New I/O write performance is **14316** MB/s
- The performance of the write, compare to the basic version, was improved almost **9.4** times
- The parameters that are required to modified are depending on the application, the resources, and the I/O pattern

Open slide master to edit

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.

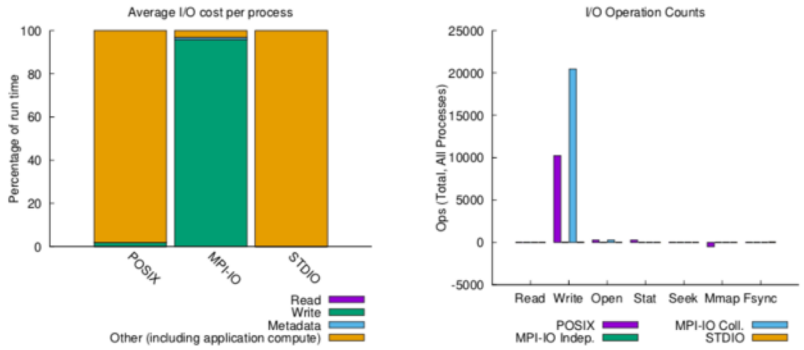
OAK RIDGE National Laboratory

Open slide master to edit

Darshan on Summit – Optimizing blocking PNetCDF

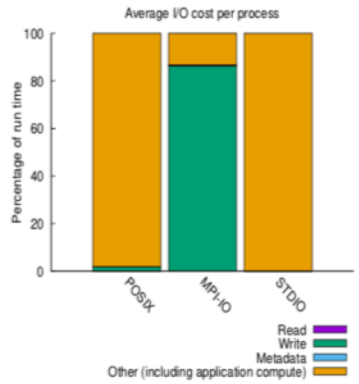
jobid: 170626	uid:	nprocs: 256	runtime: 99 seconds

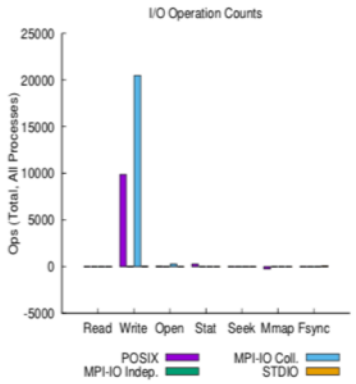
I/O performance *estimate* (at the MPI-IO layer): transferred 959 MiB at 1627.63 MiB/s
I/O performance *estimate* (at the STDIO layer): transferred 0.0 MiB at 21.65 MiB/s



jobid: 171040	uid:	nprocs: 256	runtime: 13 seconds

I/O performance *estimate* (at the MPI-IO layer): transferred 1010 MiB at 13852.43 MiB/s
I/O performance *estimate* (at the STDIO layer): transferred 0.0 MiB at 26.65 MiB/s





OAK RIDGE National Laboratory

Open slide master to edit

Conclusion – Spectrum Scale

- 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

Open slide master to edit

OAK RIDGE National Laboratory

Burst Buffer on Summit

Burst Buffer on compute node

- Burst Buffers are technologies that provide faster I/O based on new media, on Summit we have on each compute node a Samsung PM1725a NVMe
- 4,608 nodes with local NVMe of 1.6 TB
  - 7.3 PB Total
  - Write performance per BB node: 2.1 GB/s
  - Read performance per BB node : 5.5 GB/s
- By default we can do one file per MPI process or one file per node, no single shared file between different Burst Buffer nodes without using any other Burst Buffer library (check second part of the session).
- Linear scalability by using Burst Buffers across many nodes
- Exclusive usage of the resources, no sharing with other users

Burst Buffer – Use cases

- Periodic burst
- Good for machine learning and deep learning workloads
- Transfer to PFS between bursts
- I/O improvements
- Improves applications with heavy metadata

Open slide master to edit

Burst Buffer

- Burst Buffer can be used through the scheduler, integration with LSF
- What a user has to do?
  - Add the appropriate scheduler option in the submission script
  - Copy any necessary file on the Burst Buffer (input file, executable)
  - Execute the application and make sure that it reads/writes the files with significant size from Burst Buffer
  - Copy required files from Burst Buffer to Spectrum Scale

Open slide master to edit

Submission script for Burst Buffer – NAS BTIO

GPFS

```
#!/bin/bash
#BSUB -P projid
#BSUB -J nas_btio
#BSUB -o nas_btio.o%J
#BSUB -W 10
#BSUB -nnodes 1

jsrun -n 1 -a 16 -c 16 -r 1 ./btio
```

Burst Buffer

```
#!/bin/bash
#BSUB -P projid
#BSUB -J nas_btio
#BSUB -o nas_btio.o%J
#BSUB -W 10
#BSUB -alloc_flags "nvme"
#BSUB -nnodes 1

jsrun -n 1 cp btio  inputbt.data /mnt/bb/$USER/

jsrun -n 1 -a 16 -c 16 -r 1 /mnt/bb/$USER/btio

jsrun -n 1 cp 1 /mnt/bb/$USER/btio.nc
/gpfs/alpine/scratch/…
```


Open slide master to edit

NAS BTIO

- Executing 16 MPI processes on a **single** BB node, blocking PNetCDF with a single shared file

Total I/O amount	:	152.6 GB
Time in sec	:	67.98
I/O bandwidth	:	2.24 GB/s

OAK RIDGE National Laboratory
Open slide master to edit

Understanding the MPI I/O Hints

- Using the command *export ROMIO_PRINT_HINTS=1* in the submission script, we can acquire the following information for 16 MPI processes of one BB node

```
key = cb_buffer_size          value = 16777216
key = romio_cb_read             value = enable
key = romio_cb_write            value = enable
key = cb_nodes                  value = 1
…
key = cb_config_list            value = *:1
key = romio_aggregator_list     value = 0
```


OAK RIDGE National Laboratory
Open slide master to edit

NAS BTIO - Improved

- Increasing the MPI I/O aggregators to 8

  echo "cb_config_list *:8" > romio_hints

- Declare the ROMIO_HINTS variable

  export ROMIO_HINTS=$PWD/romio_hints

- New performance results

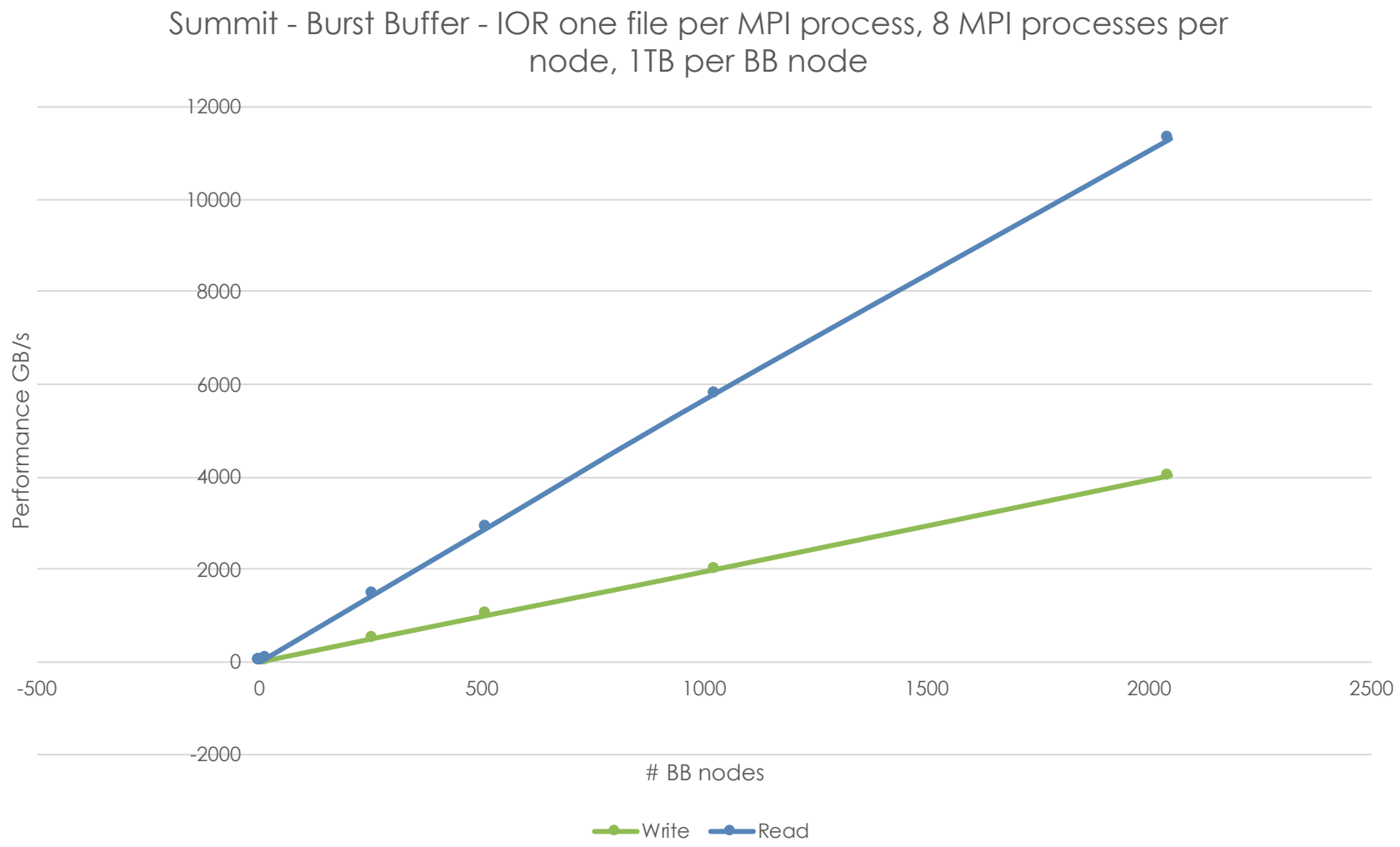
  Totail I/O amount : 152.6 GB
  Time in sec : 52.47
  I/O bandwidth : 2.98 GB/s

Almost 23% improvement by using page cache and NVMe

Open slide master to edit

Burst Buffer

- Scalability test with IOR



OAK RIDGE National Laboratory

Open slide master to edit

Conclusions – Burst Buffer

- Burst Buffer is the solution for heavy I/O applications
- We need some extra libraries on Summit to support various workflows
- Tuning with MPI I/O hints could provide faster execution time

Open slide master to edit

OAK RIDGE National Laboratory

Burst Buffer libraries

Slides from Chris Zimmer

Modes of Use

- Spectral
  - File per process checkpoints
  - Iterative output
- SymphonyFS
  - Shared file output

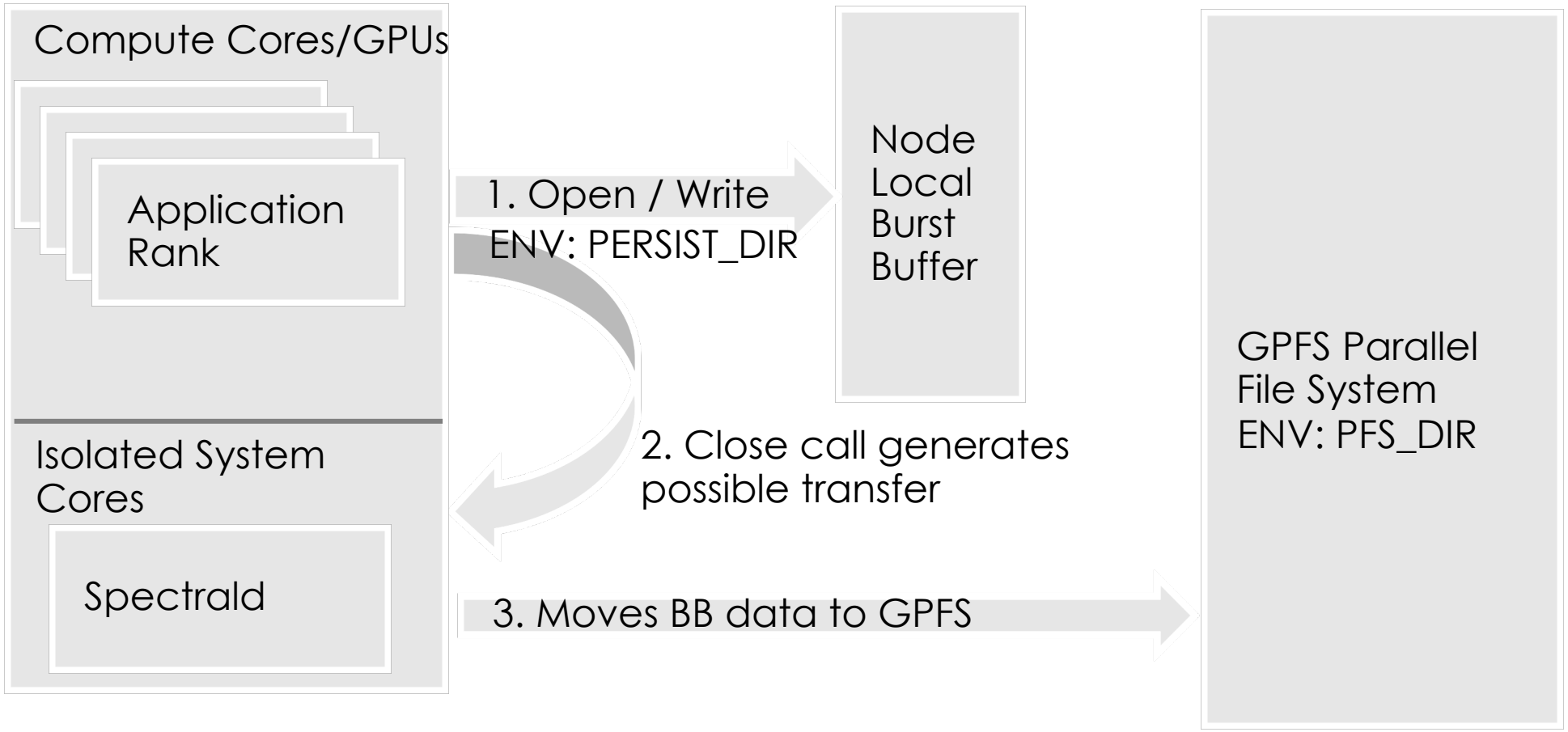
OAK RIDGE National Laboratory
Open slide master to edit

Spectral

- On node copy agent
  - Runs on isolated cores as system agent
- Application Interface Transparent
  - Node code modifications (LD_PRELOAD)
    - Changes limited job scripts
  - Application only reasons about a single namespace
- Preserves portability with single namespace
- Non-shared files

Open slide master to edit

Spectral Data Flow



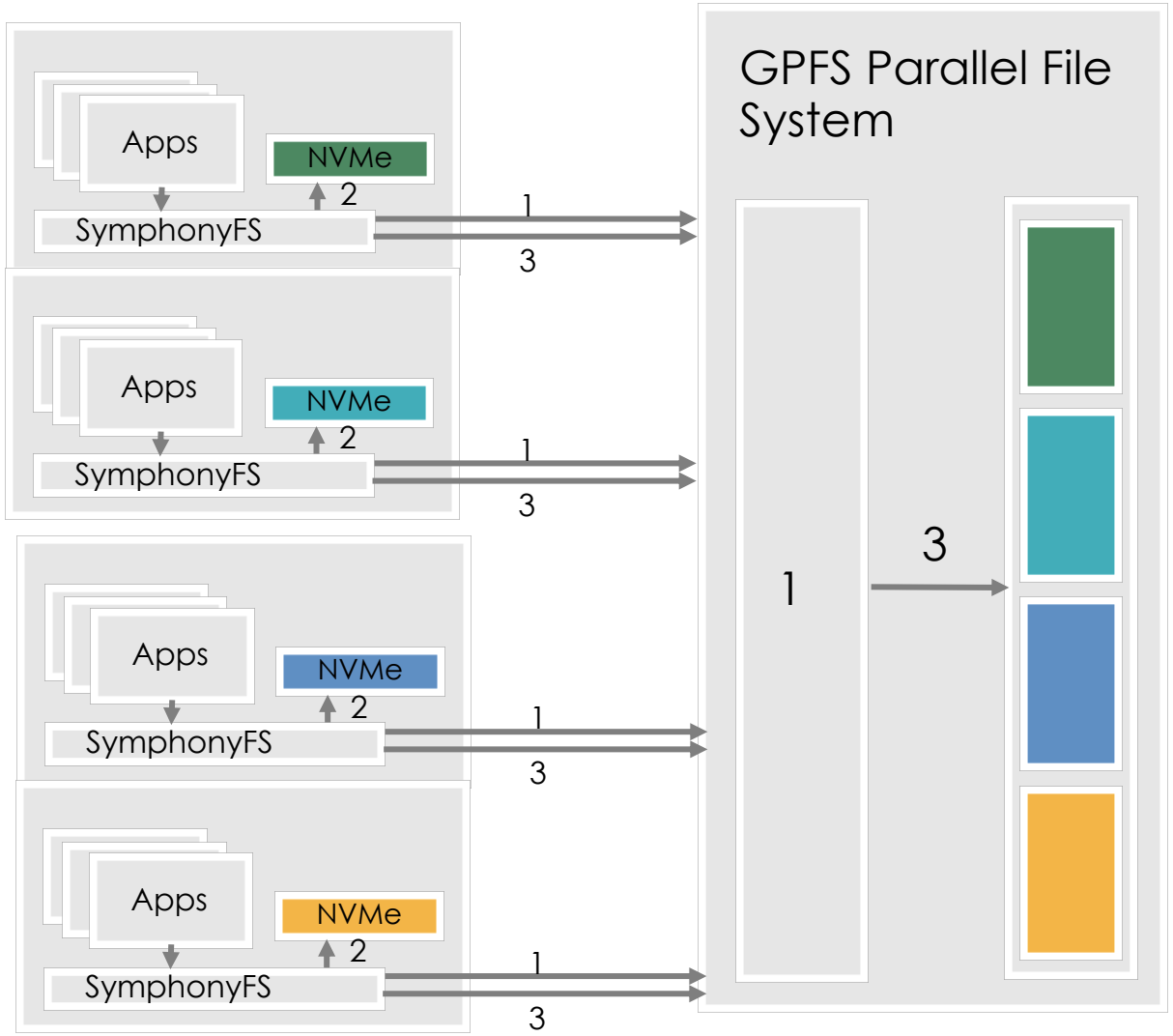
OAK RIDGE National Laboratory
Open slide master to edit

SymphonyFS

- Single (FUSE) name space
- File per process, shared file
- Operational model:
  - Uses NVMe as a write-back extent cache
  - Reconstructs file on parallel file system

Open slide master to edit

SymphonyFS Data Flow



1. File open (Meta data handled through GPFS)'
2. Apps write data (buffered into NVMe's)
3. Upon file close and flush. SymphonyFS reconstitutes file on GPFS.

OAK RIDGE National Laboratory
Open slide master to edit

SymphonyFS

- Limitations
  - Non-overlapping writes
  - Read after write
    - Must be flushed to parallel file system
      - Reads come from parallel file system

Open slide master to edit

Questions/Interest in early access?

- Spectral - zimmercj@ornl.gov

- SymphonyFS – brumgardcd@ornl.gov

OAK RIDGE National Laboratory
Open slide master to edit

BB libraries evaluation

- When the libraries are ready we'll start the evaluation and the documentation
- We need test cases, if you think that your application could benefit from BB, contact us
- SymphonyFS will be released later than Spectral library

OAK RIDGE National Laboratory
Open slide master to edit

Thank you!

Questions?

help@olcf.ornl.gov

Open slide master to edit