OAK RIDGE National Laboratory

Burst Buffer on Summit

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

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

U.S. DEPARTMENT OF ENERGY

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

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/on Burst Buffer
  - Copy the necessary files back to GPFS to save them

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 /mnt/bb/$USER/btio.nc
/gpfs/alpine/scratch/…
```

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

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
```

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

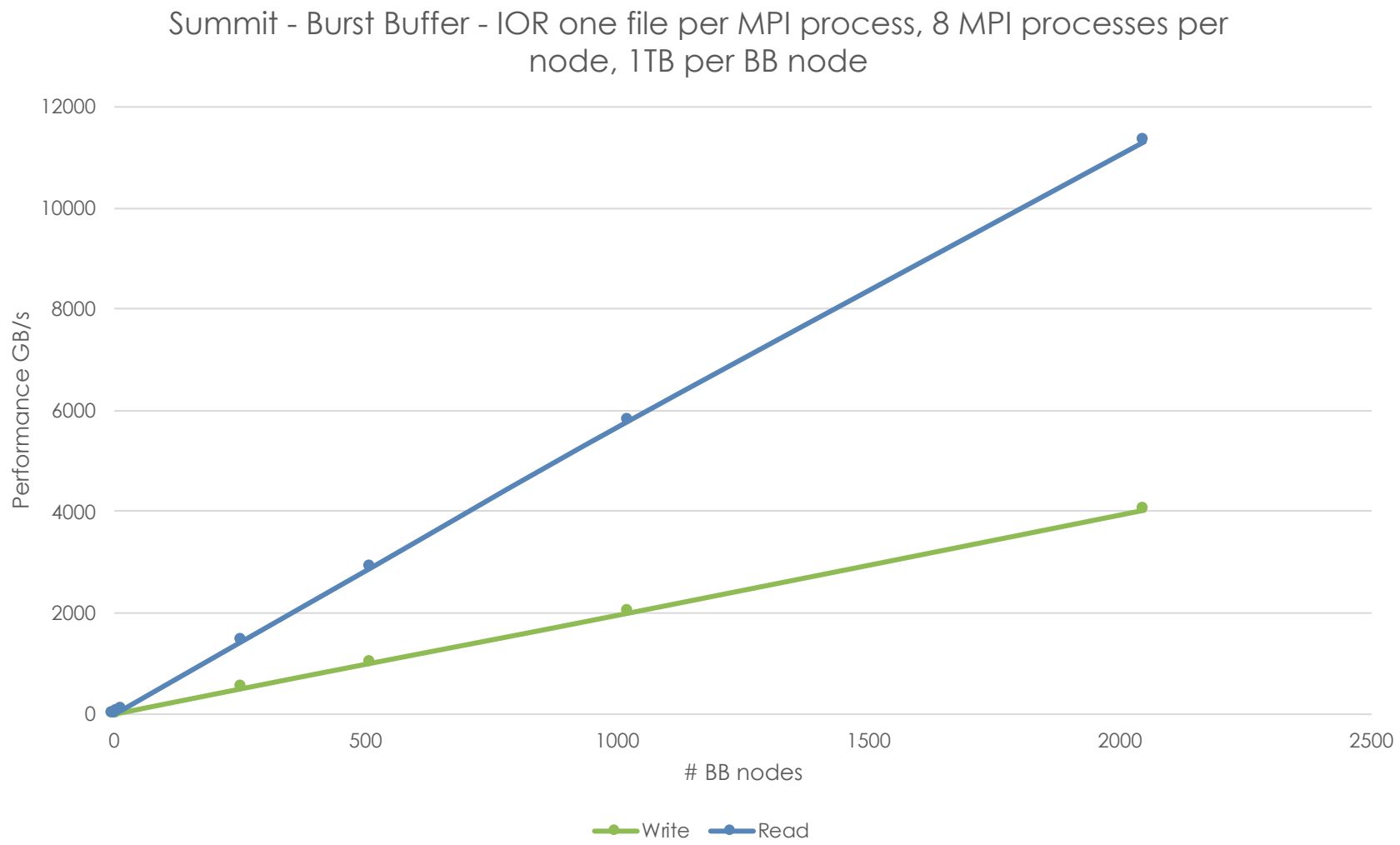
- New performance results

  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

Burst Buffer

- Scalability test with IOR



OAK RIDGE National Laboratory

Documentation

OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

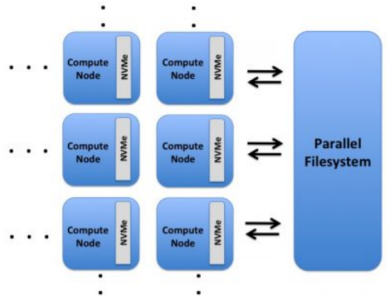
ABOUT OLCF ▾ OLCF RESOURCES ▾ R&D ACTIVITIES ▾ SCIENCE AT OLCF ▾ FOR USERS ▾ OLCF MEDIA ▾

- Connecting
- File Systems: Data Storage & Transfers
  - Storage Overview
  - User-Centric Data Storage
  - Project-Centric Data Storage
  - Data Transfer and Summit
  - HPSS Best Practices
  - Burst Buffer
- Software
- Shell & Programming Environments
- Compiling
- Running Jobs
- Debugging

Burst Buffer

NVMe (XFS)

Each compute node on Summit has a **N**on-**V**olatile **Me**mory (NVMe) storage device, colloquially known as a "Burst Buffer" with theoretical performance peak of 2.1 GB/s for writing and 5.5 GB/s for reading. Users will have access to an 1600 GB partition of each NVMe. The NVMes could be used to reduce the time that applications wait for I/O. Using an SSD drive per compute node, the burst buffer will be used to transfers data to or from the drive before the application reads a file or after it writes a file. The result will be that the application benefits from native SSD performance for a portion of its I/O requests. Users are not required to use the NVMes. Data can also be written directly to the parallel filesystem.



The NVMes on Summitdev are local to each node.

Current NVMe Usage

▲ TOP

Using GPFS	*Using NVMe*
`#!/bin/bash`	
`#BSUB -P xxx`	
`#BSUB -J NAS-BTIO`	
`#BSUB -o nasbtio.o%J`	
`#BSUB -e nasbtio.e%J`	
`#BSUB -W 10`	
`#BSUB -nnodes 1`	
	`#BSUB -alloc_flags nvme`
	`export BBPATH=/mnt/bb/$USER/`
	`jsrun -n 1 cp btio ${BBPATH}`
	`jsrun -n 1 cp input* ${BBPATH}`
`jsrun -n 1 -a 16 -c 16 -r 1 ./btio`	`jsrun -n 1 -a 16 -c 16 -r 1 ${BBPATH}/btio`
`ls -l`	`jsrun -n 1 ls -l ${BBPATH}/`
	`jsrun -n 1 cp ${BBPATH}/* .`

OAK RIDGE National Laboratory

Conclusions

- 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

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?