Experiences with the Burst Buffer at NERSC

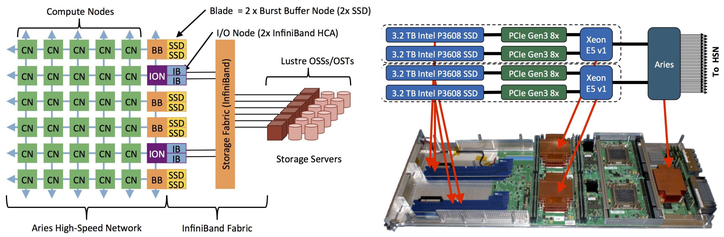
Debbie Bard, Wahid Bhimji, David Paul, Glenn K. Lockwood, Nicholas J Wright, Katie Antypas, Prabhat, NERSC, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA, Email: wbhimji@lbl.gov
Science Applications: Steve Farrell, Andrey Ovsyannikov, Melissa Romanus, Brian Van Straalen, David Trebotich, Guenter Weber

Introduction

NVRAM-based Burst Buffers are an important part of the emerging HPC storage landscape. The National Energy Research Scientific Computing Center (NERSC) at LBNL recently installed one of the first Burst Buffer systems as part of its new Cori supercomputer, collaborating with Cray on the development of the DataWarp software. NERSC has over 6500 users in 750 different projects spanning a wide variety of scientific applications, including climate modeling, combustion, fusion, astrophysics, computational biology, and many more. The applications of the Burst Buffer at NERSC are therefore also considerable and diverse.

We describe here experiences with the first year of the NERSC Burst Buffer. A number of research projects have had early access to the Burst Buffer and have exercised its capabilities to enable new scientific advancements. We present performance results and lessons-learnt from these real applications as well as benchmark results.

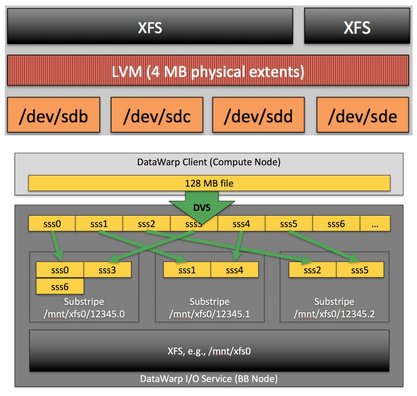
NERSC Burst Buffer Architecture



- Phase 1 System (use-cases here): 920 TB on 144 Burst Buffer Nodes
  - Phase 2 System - now installed - double this capacity (and bandwidth)
- DataWarp software (integrated with SLURM WLM) allocates portions of available storage to users per-job (or 'persistent' across multiple jobs)
- Users get an on-demand POSIX filesystem - that can be striped across multiple Burst Buffer nodes.

Software Environment

- Logical Volume Manager (LVM) group SSDs into one block device.
- An XFS file system is created for every Burst Buffer allocation.
- The DataWarp File System (DWFS), is a stacked file system that provides namespaces
- Cray Data Virtualization Service (DVS) enables communication between DWFS and the compute nodes.
- File written from compute node ends up as (configurable) 8MB chunks, laid out across the three (configurable) substripes on the Burst Buffer node.
- Users can specify data to be staged in/out from Lustre FS while job is in queue
- Users can interact via SLURM batch script (example right) or via DataWarp API.




```
#!/bin/bash
#SBATCH -p regular -N 10 -t 00:10:00
#DW jobdw capacity=1000GB access_mode=striped type=scratch
#DW stage_in source=/lustre/file.dat \
           destination=$DW_JOB_STRIPED/ type=file
#DW stage_out source=$DW_JOB_STRIPED/output \
                destination=/lustre/output    type=directory
srun my.x --infile=$DW_JOB_STRIPED/file.dat \
                --outdir=$DW_JOB_STRIPED/output
```

Science Use Cases

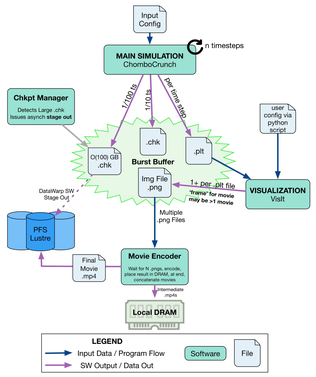
The NERSC Burst Buffer Early User program supported around 30 applications to adapt code and workflows to use the Burst Buffer. These covered a variety of use cases, including read or write-intensive I/O patterns, workflow coupling and high IOPs. We highlight two projects below, for many more studies see [1],[2].

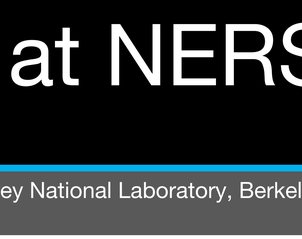
Coupled science workflow using the Burst Buffer: ChomboCrunch + VisIT

ChomboCrunch is a simulator for carbon sequestration that models fluid dynamics in ground layers down to very small physical scales. Higher resolution simulation gives higher fidelity results, but increases size of output data files (up to 100s TB). The resulting IO bottleneck can limit the scientific use of the simulation.

VisIT is a visualisation and analysis tool for scientific data. It reads the simulation data files and produces images for encoding into movie for analysis.

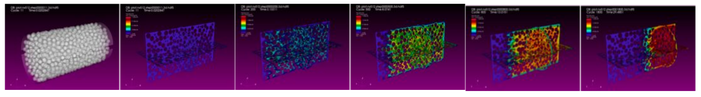
Both tasks can run simultaneously using the Burst Buffer, enabling higher spatial and temporal resolution simulations. A subset of data files, and regular checkpoint data, can be bled out to long-term storage in-place.





Results:

- Burst Buffer significantly out-performs Lustre at all resolution levels.
- Bandwidth scales as expected as resolution (and therefore output data size) scales up.
- Large-scale simulation:
  - 'Packed cylinder'[3] on 8192 cores over 256 nodes with 8 further nodes used for VisIt.
  - Full BB, 140 nodes: >90GB/s obtained.
- Below: snapshots of resulting VisIT movie of fluid flow through soil cylinder.
- Compute node/BB node scaled from 16/1 to 1024/ 64
- Lustre results used a 1MB stripe size and a stripe count of 72 OSTs



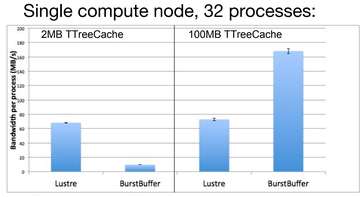
Enabling experimental data analysis on HPC with the Burst Buffer: ATLAS LHC experiment

The ATLAS Large Hadron Collider experiment produces 100s PB /year of data. This is processed worldwide mainly on conventional compute clusters. HPC machines are currently used only for simulations. The Burst Buffer can enable other workloads such as analysis of data (simulated and experimental) on HPC, which can have challenging I/O patterns.

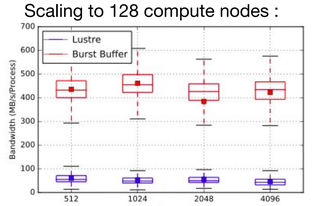
Results:

- Poor initial Burst Buffer I/O performance compared to Lustre.
- Increase application memory cache to 100M means less reads so 17x boost in I/O bandwidth on Burst Buffer which then outperforms Lustre.
- Scales well to larger job sizes:
  - 50TB dataset: 143 node Burst Buffer allocation
  - 8x speedup in I/O time relative to Lustre
  - IO not a bottleneck for this workflow on Cori

Single compute node, 32 processes:



Scaling to 128 compute nodes :



System Monitoring

Benchmarks

On installation of Cori Phase 1, the IOR Benchmark was run on 1120 Compute Nodes with 4 ranks/node and using 140 Burst Buffer Nodes to obtain the peak results shown below. Bandwidth tests used 8 GB block-size and 1MB transfers. IOPS tests used 1M blocks and 4k transfers. MPIO shared file performance is improved in later versions of the Cray Data Warp software which will be used for the Phase 2 system.

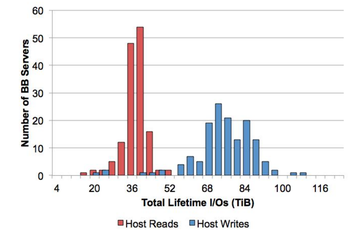
IOR Posix FPP		IOR MPIO Shared File		IOPS	
Read	Write	Read	Write	Read	Write
905 GB/s	873 GB/s	803 GB/s	351 GB/s	12.6 M	12.5 M

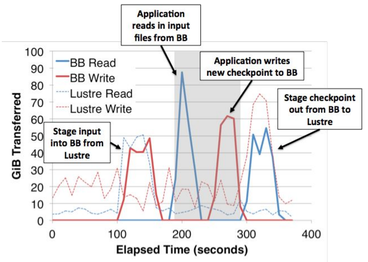
Usage Monitoring

NERSC is collecting system-level monitoring information related to the Burst Buffer through various routes. This complements application-level metrics shown in the use-case section.
For example:

1. Intel SSD Data Centre tool for NVMe. This allows device-level I/O monitoring. Using this, the plot top-right shows the balance of total I/O across Burst Buffer nodes and that, during this early-user period, the workload was write heavy.

2. Collectd on Burst Buffer nodes. This allows node-level I/O monitoring which can be correlated with data from the Lustre LMT monitoring to track how an application's I/O translates to the system. This is shown for a checkpoint-restart workflow in the plot bottom-right.





Conclusions

NERSC has successfully brought a Burst Buffer into production with its new Cori system. This offers a novel approach to creating flexibly-sized, on-demand filesystems backed by high-performance NVRAM hardware. The Phase 1 system is capable of around 900 GB/s bandwidth and 12.5M IOPs. We ran an Early User Program which was crucial to our debugging of this complex new technology. It exposed issues and led to performance improvements as well as fixes to scaling limits, operational problems and usability issues; impossible to achieve with synthetic tests or with large numbers of generic users. We have also put in place system level monitoring to further diagnose and tune performance.

Through these efforts and thanks also to development by Cray and SchedMD, the NERSC Burst Buffer now functions well in production. We highlight here a couple of Burst Buffer science use-cases: coupling of simulation and visualization with Chombo-Crunch and VisIT, and accelerating analytics on data from the ATLAS LHC experiment.

In conclusion the NERSC Burst Buffer provides a high-performance solution for scientific I/O and is now starting to enable new science workflows.

References:
[1] Bhimji, W. *et. al.*, "Accelerating Science with the NERSC Burst Buffer Early User Program", *Proceedings of Cray Users Group* https://cug.org/proceedings/cug2016_proceedings/includes/files/pap162.pdf
[2] Bhimji, W. *et. al.*, "Extreme I/O on HPC for HEP using the Burst Buffer at NERSC", *Computing in High-Energy Physics (CHEP) 2016*, https://indico.cern.ch/event/505613/contributions/2227423/
[3] D. Trebotich and D. Graves, "An adaptive finite volume method for the incompressible navier–stokes equations in complex geometries," *Communications in Applied Mathematics and Computational Science*, vol. 10, no. 1, pp. 43–82, 2015.