NERSC Burst Buffer

Presentation By: Justin Shafner

Presented On: 4/13/2021

“Accelerating Science with the NERSC Burst Buffer Early User Program”

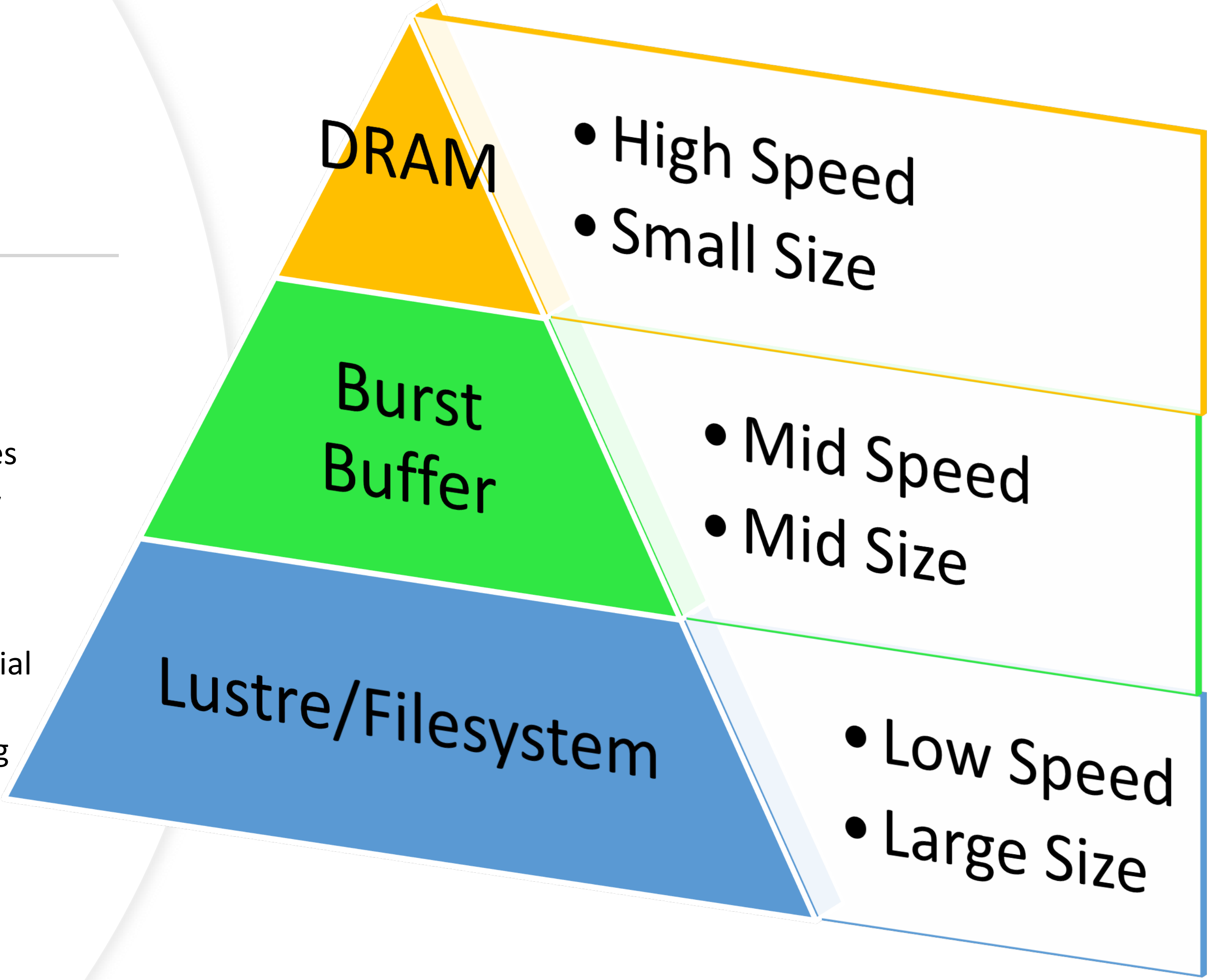
Wahid Bhimji*, Debbie Bard*, Melissa Romanus*†, David Paul*,
Andrey Ovsyannikov*, Brian Friesen*, Matt Bryson*, Joaquin Correa*,
Glenn K. Lockwood*, Vakho Tsulaia*, Suren Byna*, Steve Farrell*, Doga Gursoy‡,
Chris Daley*, Vince Beckner*, Brian Van Straalen*, David Trebotich*, Craig Tull*, Gunther Weber*,
Nicholas J. Wright*, Katie Antypas*, Prabhat*
* Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA, Email: wbhimji@lbl.gov
† Rutgers Discovery Informatics Institute, Rutgers University, Piscataway, NJ, USA
‡Advanced Photon Source, Argonne National Laboratory, 9700 South Cass Avenue, Lemont, IL 60439, USA

Overview:

- The function and use of a Burst Buffer
- How it is implemented in hardware
- How it is implemented in software
- Performance benchmarks
- Five science project test cases
- Lessons learned and conclusion

Burst Buffer:

- Mid-Level storage solution
- Comprised of high-speed SSDs
  - Connected to network not compute nodes
  - Controlled by DataWarp software by Cray
- Common Targeted Applications
  - Checkpointing and Restarting
  - Complex I/O patters such as non-sequential table lookups
  - Coupling applications and post-processing
  - Dynamic Load Balancing
  - Interactive visualization workflow



Cori Supercomputer System: Burst Buffer Hardware

- Phase 1 “Cori Data Partition”
  - Cray XC40 w/ 1632 dual-socket nodes
  - Two 2.3 GHz 16-core Haswell
  - 128 DRAM per node
  - Dragonfly topology
  - Burst Buffer nodes with SLURM
  - High performance Lustre scratch w/ 27 PB at 700 GB/s peak

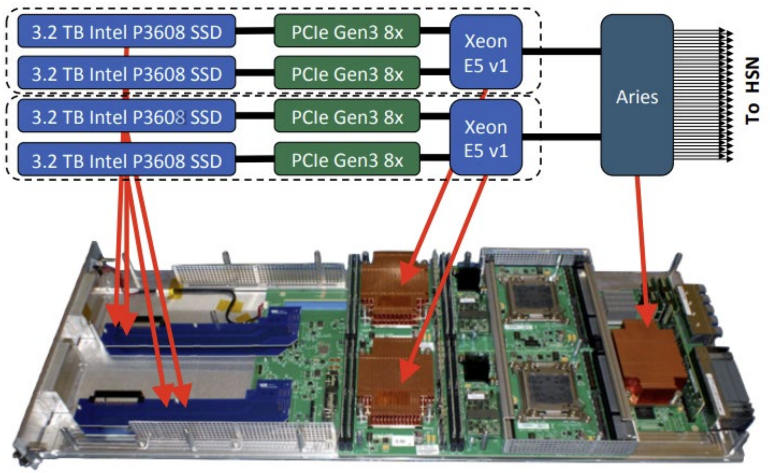


Fig. 1. A Cori Burst Buffer Blade

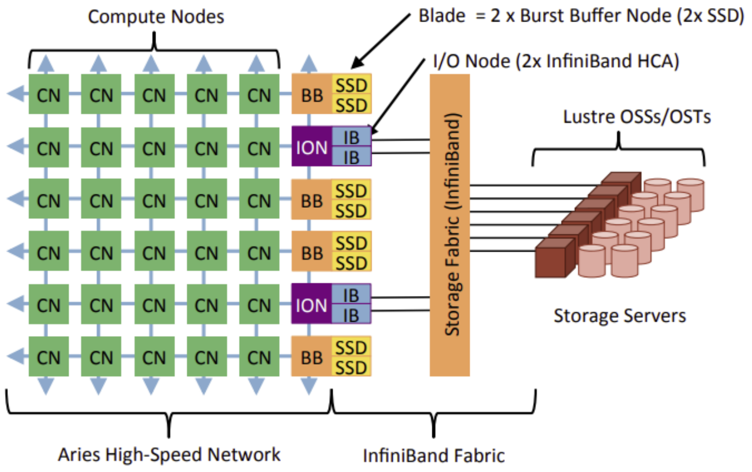


Fig. 2. Placement of Burst Buffer nodes in the Cori System

Burst Buffer Software Environment:

- DataWarp by Cray
- Logical Volume Manager (LVM)
  - Groups multiple SSDs on a single node into one logical block device
- XFS File System
  - Created for each Burst Buffer Allocation
  - This allocation appears as separate filesystem to the user
- DataWarp File System (DWFS)
  - Based on wapfs
  - Coordinates communication between BB nodes
- Cray Data Virtualization Service (DVS)
  - Handles communication between compute nodes and DWFS

To illustrate the normal user-level interaction with the Burst Buffer, example batch script directives are given below:

```
#DW jobdw capacity=1000GB \
  access_mode=striped type=scratch
#DW stage_in source=/lustre/inputs \
  destination=$DW_JOB_STRIPED/inputs \
  type=directory
#DW stage_in source=/lustre/file.dat \
  destination=$DW_JOB_STRIPED/ type=file

#DW stage_out source=$DW_JOB_STRIPED/outputs \
  destination=/lustre/outputs type=directory
srun my.x --indir=$DW_JOB_STRIPED/inputs \
  --infile=$DW_JOB_STRIPED/file.dat  \
  --outdir=$DW_JOB_STRIPED/outputs
```

Fig. 3. Schematic of the different layers of the DataWarp software. The Burst Buffer node is logically 4 SSD block devices, which are aggregated by LVM. When a Burst Buffer allocation is requested by the user, an XFS filesystem is created by the DataWarp software on that allocation. In this example, two different allocations coexist on the same Burst Buffer node

Benchmark Performance:

- 8 Gb block size and 1MB transfer size
- Used file sizes 1.5x the compute node's memory
- 1120 compute nodes with 4 processes per node
- 140 Burst Buffer Nodes
- Compared to Lustre
  - 700-750 GB/s POSIX File-Per-Process
  - 573/223 GB/s Read/Write for MPI-IO

TABLE I
I/O PERFORMANCE TO THE CORI BURST BUFFER AS MEASURED WITH THE IOR BENCHMARK (DETAILS GIVEN IN THE TEXT)

Posix File-Per-Process		MPI-IO 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

Nyx/Boxlib:

- Cosmology AMR simulating dark matter
- Simulates from 900,000 years post Big-Bang to present day
- Data files O(TB) with even larger checkpoints
- 256x256x256 average mesh points decomposed into 1.8 GB sub-volumes
- Peak theoretical bandwidth achieved when using 16 writing processes (higher than Lustre)

IV. Science Use Cases

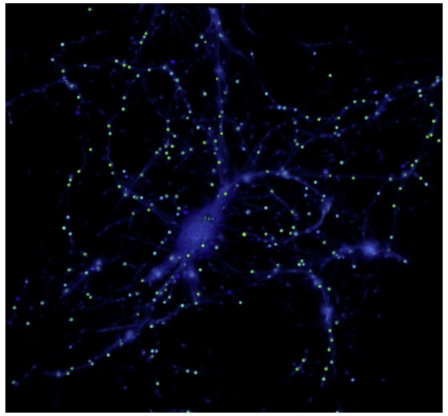
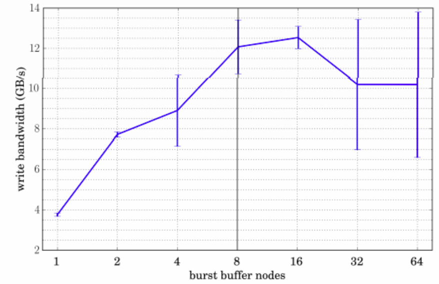
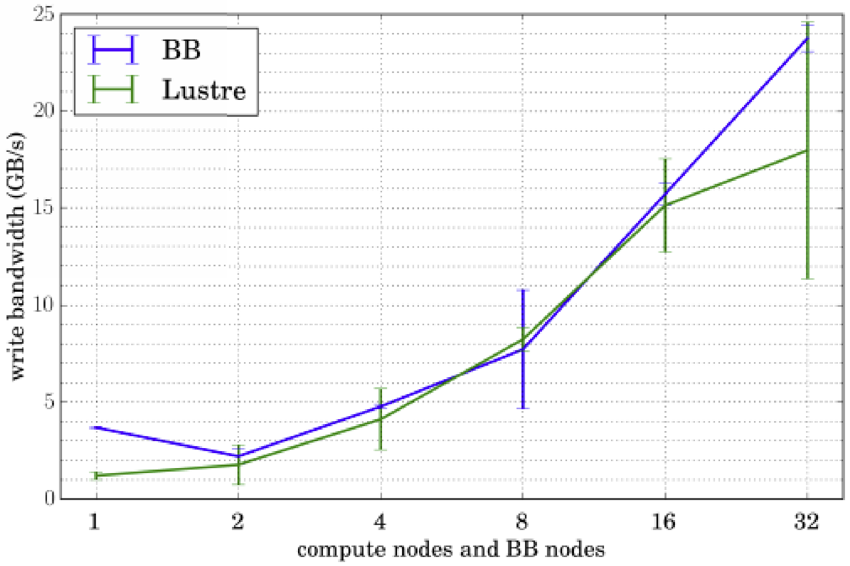


Fig. 5. Volume rendering of a Nyx calculation at redshift z=3. The problem domain spans 10 Mpc per side, discretized on a grid of size 512^3 points. The baryonic matter collapses into filaments, shown in blue. The green spheres indicate the centroids of dark matter halos which surround the filaments.



(a) IO performance of Nyx for different numbers of Burst Buffer nodes, for a simulation using 8 compute nodes, each with 2 MPI writers. The vertical black line denotes 1 Burst Buffer node for each compute node.



Chombo-Crunch:

- Carbon Dioxide injection into soil
- Non-linear dynamic multi-phase flow modeling
- Fluid-fluid interactions and coupled fluid-solid
- Output files range from O(10) GB to O(1) TB
- Realtime I/O creating PNG and Movie files interactively
- Too intensive for disk-based storage

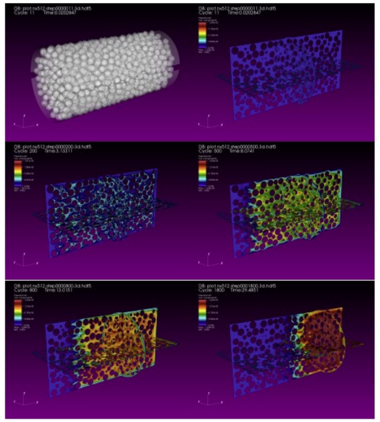


Fig. 10. Snapshots from movie produced by a 'packed cylinder' [14] Chombo-Crunch and VisIt workflow. Full movie is available from http://portal.nersc.gov/project/mpccc/wbhimji/BurstBuffer/packedCylinder.mp4

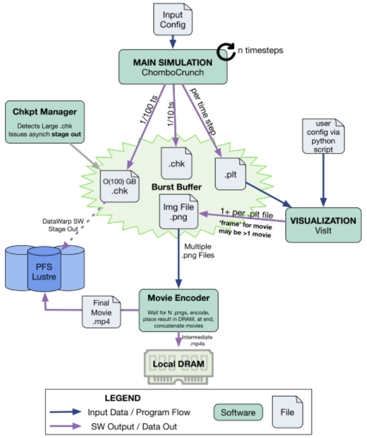
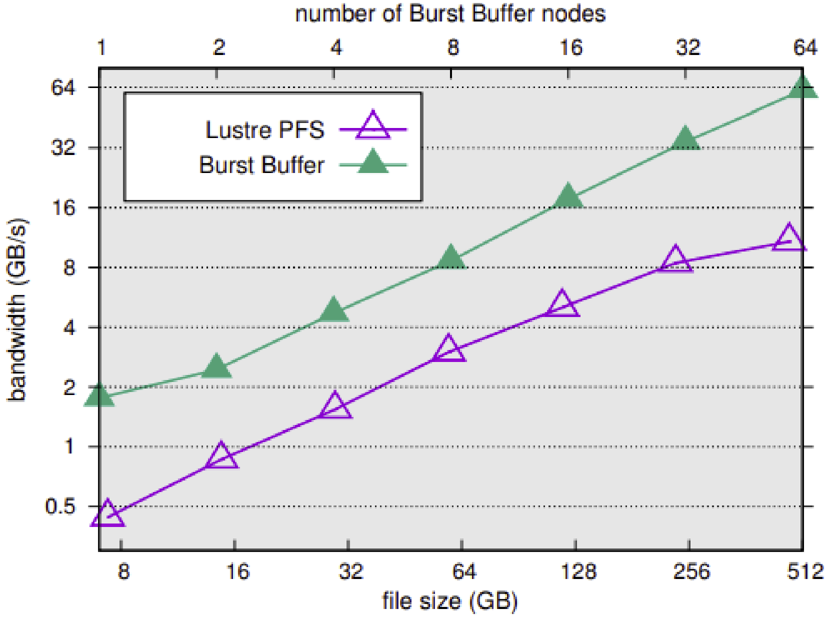


Fig. 8. Chombo-Crunch + VisIt workflow



Chombo-Crunch:

VPIC-IO:

- Plasma simulation of charged particles
- Simulates mapping of magnetic field in interstellar medium
- Simulates trillions of particles ranging from 32 – 40 TB data files per time step
- Generates O(10,000) time steps
- Fast store to BB then asynchronously move to Lustre storage
- IO optimized by having the same number of BB nodes as MPI aggregators
- Independent I/O outperforms since SSDs perform better at random value lookup operations

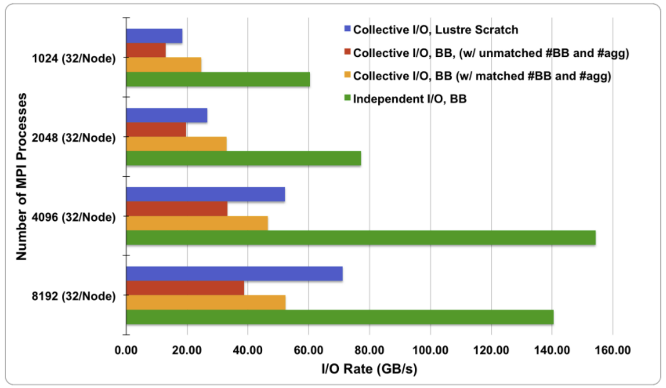


Fig. 12. Performance of the VPIC-IO kernel, comparing different scales and configurations of MPI-IO. Blue: I/O bandwidth for MPI collective I/O using the Lustre PFS; Red: MPI collective I/O using the Burst Buffer and the number of aggregators not matching the number of Burst Buffer nodes; Yellow: matching the number of MPI aggregators; Green: using independent I/O on the Burst Buffer.

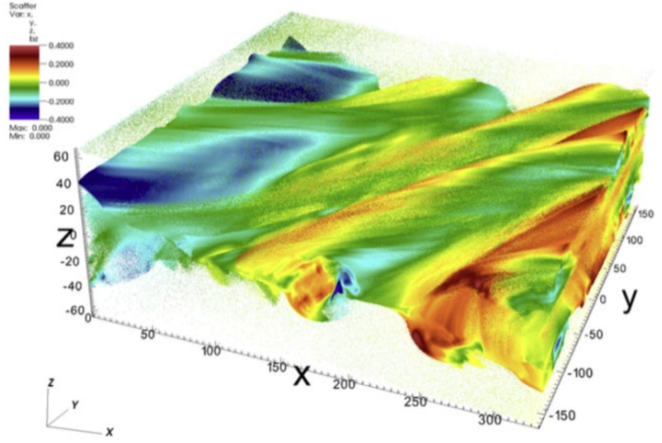


Fig. 11. Mapped magnetic field data in the z dimension (bz) for all highly energetic particles (Energy>1.5) in physical space (x, y, and z dimensions). The plots use a linear color scale. The range is restricted to [0.4, 0.4] for bz.

TomoPy ALS Spot Suite

- Tomographic Reconstruction (CT scans)
- Advanced Light Source (ALS) and Advanced Photon Source (APS) augment raw 2D images from scanners
- Tomographic Reconstruction process is I/O intensive with 30% of the application spent in I/O

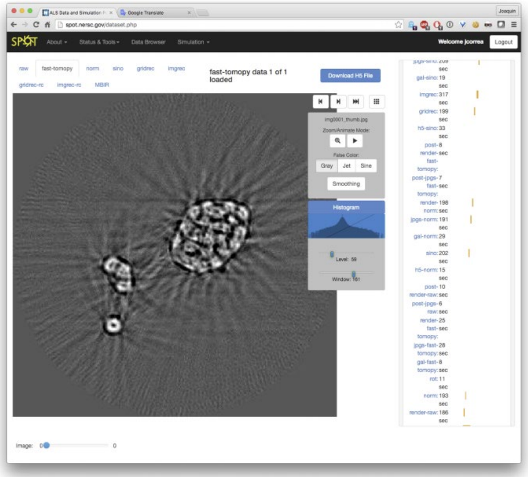


Fig. 15. A screenshot of the SPOT webportal at NERSC, illustrating both the output of TomoPy and how it is running in the production pipeline using the Burst Buffer.

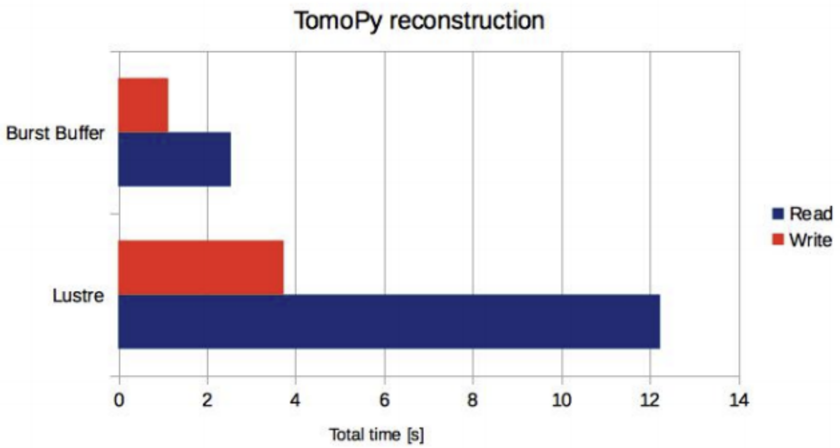


Fig. 14. Total time for read and write operations by TomoPy running on 9.9 GB of input data on 500 GB Burst Buffer allocation.

ATLAS at LHC

- ATLAS at Large Hadron Collider
- Collides protons 40 million times per second
- O(1) PB of initial data
- I/O had originally bottlenecked certain simulations
- Limited performance improvement for small cache or small file sizes

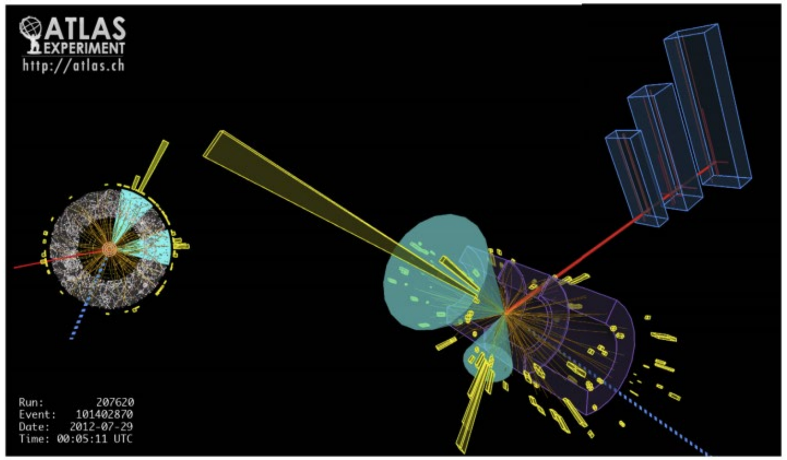


Fig. 16. Example Higgs (decaying to b-quarks) signal 'event' in the ATLAS detector.

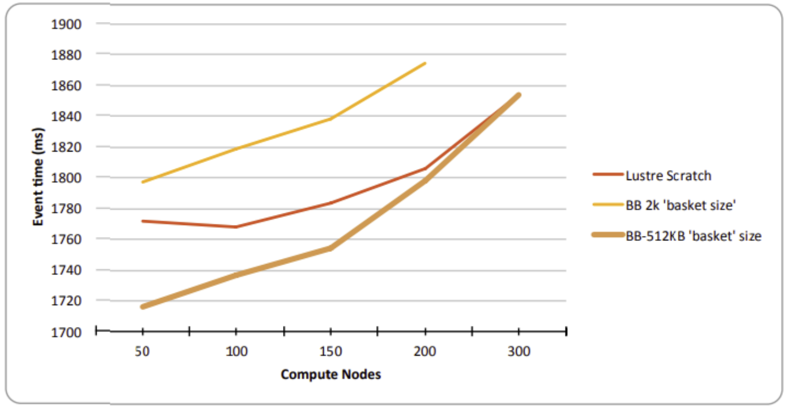


Fig. 17. Scaling of total time per 'event' for ATLAS simulation for the Lustre Scratch filesystem and the Burst Buffer with both the default 2k 'basket' (transfer) size and a 512k basket size. This time also includes compute time but this is constant for the different storage backends.

Conclusions and Lessons Learned

- Early Use programs help in debugging production technologies
- Burst Buffer provides high-performance I/O as intended with demonstrated benefits
- Small files and varied I/O patterns have poor performance
- Using MPI-IO with DataWarp needs to be optimized
- Even checkpointing to Burst Buffer requires moderate code-tuning compared to Lustre
- Achieved near theoretical-peak performance on Early Use programs (Nyx)

Questions?