Burst Buffer at KAUST

Bilel Hadri
KAUST Supercomputing Laboratory

2017
CUG.2017.CAFFEINATED COMPUTING
Redmond, Washington May 7-11, 2017

Burst Buffer Tutorial

KAUST Supercomputing Laboratory (KSL)

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- From its start in 2009, KAUST has offered HPC resources and facilities as a key technology enabler of research and discovery for multidisciplinary scientific fields.
  - Shaheen 1 BG/P - 16 racks 190.9 TF (#14 TOP500 June 2009)
  - Shaheen 2 XC40 - 36 cabinets 5.53 PF (#7 TOP500 June 2015)

- KSL provides a wide range of advanced HPC services to the on-campus research community (more than 30% of faculty members at KAUST use HPC simulation).

- Our mission is to inspire and enable scientific discoveries through development and deployment of HPC solutions, training of HPC end-users, and outreach to regional academia, industries, government agencies and beyond.

Shaheen Supercomputer

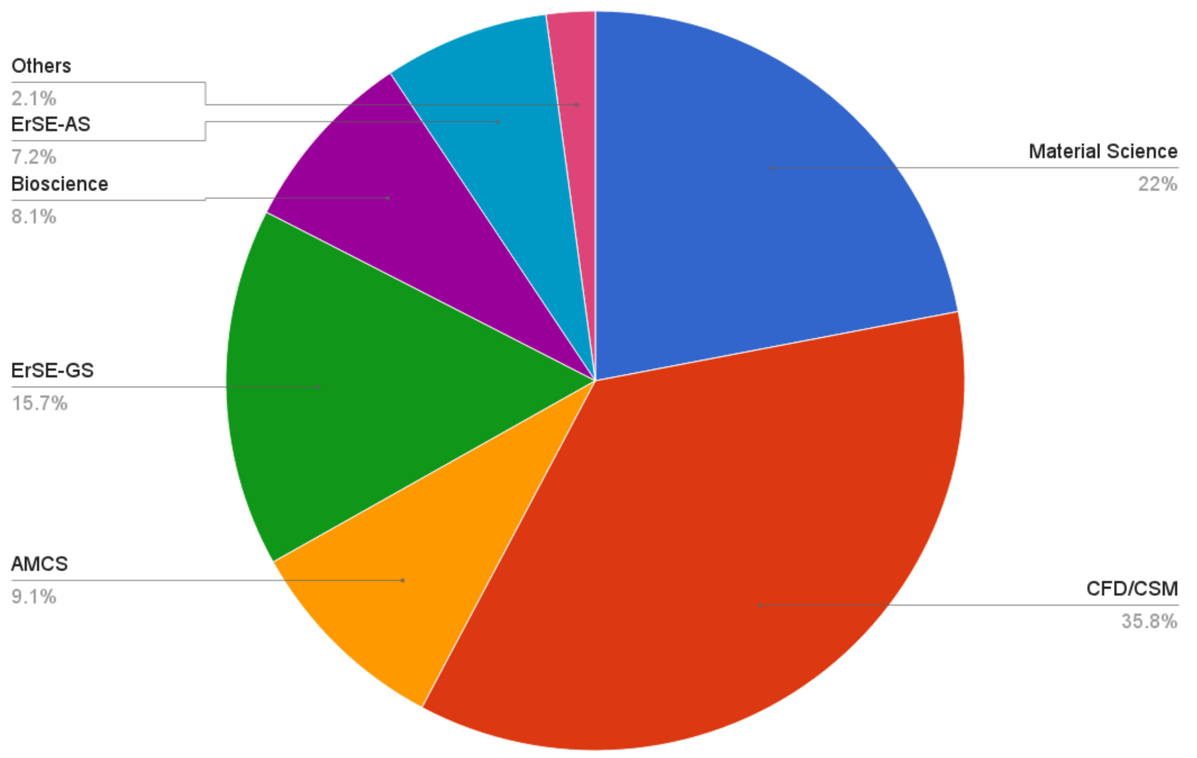
COMPUTE	Node	Processor type: Intel Haswell	2 CPU sockets per node, 16 processors cores per CPU,2.3GHz
		6174 Nodes	197,568 cores
		128 GB of memory per node	Over 790 TB total memory
	Power	Up to 3.1 MW	Water Cooled
	Weight/ Size	More than 100 metrics tons	36 Cray XC40 Compute cabinets, plus disk, blowers, management , etc..
	Speed	7.2 Pflop/s speak theoretical performance	5.53 Pflop/s sustained LINPACK and ranked 7th in July 2015 Top500 list
	Network	Cray Aries interconnect with Dragonfly topology	57% of the maximum global bandwidth between the 18 groups of two cabinets.
STORE	Storage	Sonexion 2000 Lustre appliance	17.6 Petabytes of usable storage. Over 500 GB/s bandwidth
	Burst Buffer	DataWarp	Intel Solid Sate Devices (SSD) fast data cache. 1.5 Petabytes of capacity Over 1.5 TB/s bandwidth.
	Archive	Tiered Adaptive Storage (TAS)	Hierarchical storage with 200 TB disk cache and 20 PB of tape storage, using a spectra logic tape library. (Upgradable to 100 PB)

Core Hours Usage on Shaheen2

Services: Users and Projects

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Field of Science	CPU hours	% overall
Material Science	267,222,123	22.00%
CFD/CSM	434,578,361	35.77%
AMCS	110,312,195	9.08%
ErSE-GS	190,519,231	15.68%
Bioscience	98,446,978	8.10%
ErSE-AS	87,788,622	7.23%
Others	26,000,529	2.14%



More than 1.2 Billion Core hours in the last 18 months.

HPC systems and I/O

- "A supercomputer is a device for converting a CPU-bound problem into an I/O bound problem." [Ken Batcher]

- Machines consist of three main components:
  - Compute nodes
  - High-speed interconnect
  - I/O infrastructure

- Most optimization work on HPC applications is carried out on
  - Single node performance
  - Network performance (communication)
  - I/O only when it becomes a real problem

Why do we need more efficient I/O?

- I/O subsystems are typically very slow compared to other parts of a supercomputer
  - You can easily saturate the bandwidth
- Once the bandwidth is saturated scaling in I/O stops
  - Adding more compute nodes increases aggregate memory bandwidth and flops/s, but not I/O
- Imagine a 24 hour simulation on 16 cores.
  - 1% of run time is serial I/O.
  - You get the compute part of your code to scale to 1024 cores.
  - 64x speedup in compute: I/O is 39% of run time (22’16” in computation and 14’24” in I/O).
- Efficient I/O is needed to
  - Spend more time doing science
  - Not waste resources
  - Prevent affecting other users

Why Burst Buffer?

- Primary usage model for the Burst Buffer for application acceleration for parallel I/O along with checkpoint and restart mechanism:
  - CFD/Combustion: KARFS(KAUST Adaptive Reacting Flows Solver), S3D,NGA
  - Climate: WRF, MITgcm
- New workload with new users
  - Detecting I/O usage with Darshan (enabled by default to users)
  - Bioinformatics, weather forecasting, deeplearning/analytics
    - Writing O(PetaByte) and up O(millions) of files
    - Will need only O(TeraByte) and few files after analysis
    - Not best practices for Lustre

Burst Buffer on Shaheen

- 268 DataWarp nodes integrated within the Aries Interconnect
  - Spread across the 36 cabinets
  - Each DataWarp node includes two 4TB Intel SSDs P3608 series
  - Each Intel P3600 Series SSD PCIe card internally contains 2 SSD controllers and associated SSD memory
  - 536 SSD cards in the system nodes on Shaheen II in total.
  - Total of 1.52 PiB of storage capacity with granularity 397.44GiB
    - Note that GiB is a power of 2 Unit and GB is a power of 10 Unit.

```
~> dwstat most
    pool units quantity    free      gran
wlm_pool bytes  1.52PiB 1.52PiB 397.44GiB

~> dwstat nodes
 node     pool online drain  gran capacity insts activs
nid00002 wlm_pool   true false 16MiB  5.82TiB     1      0
:
nid07618 wlm_pool   true false 16MiB  5.82TiB     1      0
```

Burst Buffer Performance on Shaheen

```
Max Write: 1468037.22 MiB/sec (1539348.60 MB/sec)
Max Read:  1581428.37 MiB/sec (1658248.05 MB/sec)

Run finished: Mon Nov  9 16:39:44 2015
fekia@cdl3:/project/k01/BBSAT/BBSAT_2.1>
```

http://www.cray.com/blog/getting-warp-speed-for-io/

- With the support of the KSL team, Cray's Joe Glenski and the Cray performance team, the IOR benchmark was launched on Shaheen using all 268 DataWarp accelerator nodes and 5,628 compute nodes and achieving 1.54 TB/s and 1.66 TB/s in IOR write and IOR read, respectively.

- Used around 200TB of SSD capacity. The run configuration used 2 MPI processes per node, 512K transfer size and each process was writing a 128GiB file and then reading it back.

- About Three time the performance of Lustre (500 GB/s)

Before running on BB

- Know your IO pattern !
  - Use profile and characterization tools
    - CrayPat, Darshan
- Optimize your runs for Lustre before !
  - Unless you want to show 200x speedup :D
  - Will help stage in/out
- Don’t forget the best practices
  - Stripping
  - Stripping
  - Stripping

I/O Utility: Darshan

- Darshan is “a scalable HPC I/O characterization tool… designed to capture an accurate picture of application I/O behavior… with minimum overhead”
- I/O Characterization
  - Sheds light on the intricacies of an application’s I/O
  - Useful for application I/O debugging
  - Pinpointing causes of extremes
  - Analyzing/tuning hardware for optimizations
- Installed by default on Shaheen
  - Requires no code modification (only re-linking)
  - Small memory footprint, no-verhead
  - Includes a job summary tool
    - Location of the gz:$DARSHAN_LOGPATH/YYYY/MM/DD/ username_exe_jobid_xxx.gz

MPI I/O hints

- The MPICH_MPIIO_HINTS variable specifies *hints* to the MPI-IO library that can, for instance, override the built-in heuristic and force collective buffering on:
- setenv MPICH_MPIIO_HINTS="*:romio_cb_write=enable:romio_ds_write=disable"
  - Placing this command in your batch file before calling your executable will cause your program to use these hints.
  - The * indicates that the hint applies to any file opened by MPI-IO,
- MPICH_MPIIO_HINTS_DISPLAY=1 will dump a summary of the current MPI-IO hints to stderr each time a file is opened.
  - Useful for debugging and as a sanity check against spelling errors in your hints.
- Full list and description of MPI-IO hint is available from the intro_mpi man page.

I/O Best Practices

- Read small, shared files from a single task
  - Instead of reading a small file from every task, it is advisable to read the entire file from one task and broadcast the contents to all other tasks.
- Limit the number of files within a single directory
  - Incorporate additional directory structure
  - Set the Lustre stripe count of such directories which contain many small files to 1. (default on Shaheen)
- Place small files on single OSTs
  - If only one process will read/write the file and the amount of data in the file is small (< 1 MB to 1 GB) , performance will be improved by limiting the file to a single OST on creation.

  → This can be done as shown below by: # lfs setstripe PathName -s 1m -i -1 -c 1 (default on Shaheen)

I/O Best Practices (2)

- **Place directories containing many small files on single OSTs**
  - If you are going to create many small files in a single directory, greater efficiency will be achieved if you have the directory default to 1 OST on creation

→# lfs setstripe DirPathName -s 1m -i -1 -c 1 (default on Shaheen)

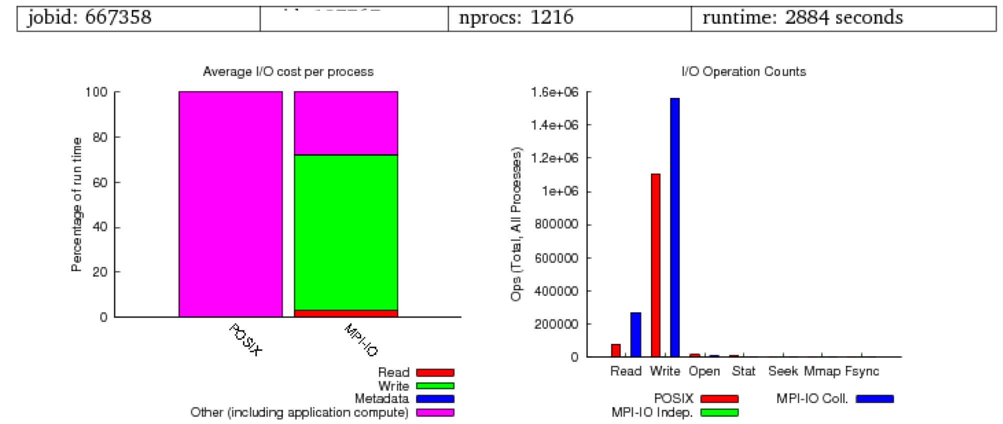
- **Avoid opening and closing files frequently**
  - Excessive overhead is created.

- **Consider available I/O middleware libraries**
  - For large scale applications that are going to share large amounts of data, one way to improve performance is to use a middleware library; such as ADIOS, HDF5, or MPI-IO

Combustion code 2x speedup

Stripe count	1	2	4	5	10
time I/O	79	48	37	42	39
time code	122	91	83	87	85
%I/O	65%	53%	45%	48%	46%
Speedup IO	1.00	1.65	2.14	1.88	2.03
Speedup code	1.00	1.34	1.47	1.40	1.44

WRF 12x speedup with file striping

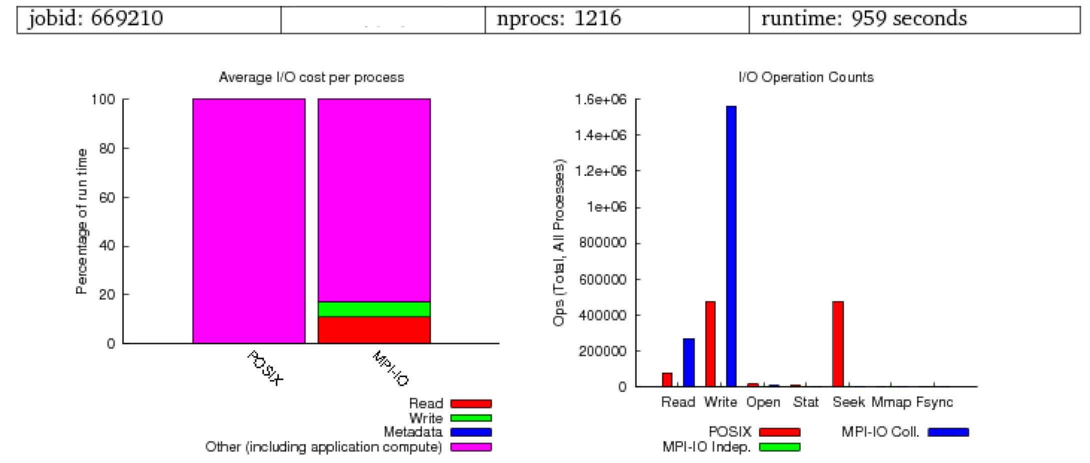


jobid: 667358		nprocs: 1216	runtime: 2884 seconds

Default striping 1,
I/O time: 2094 sec
Total time: 2884 sec

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

type	number of files	avg. size	max size
total opened	2446	485M	411G
read-only files	3	27G	77G
write-only files	6	180G	411G
read/write files	0	0	0
created files	6	180G	411G



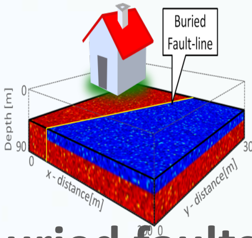
jobid: 669210		nprocs: 1216	runtime: 959 seconds

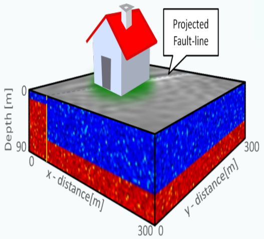
Stripping over 144 I/O time: 174 sec
Total time: 959 sec

I/O speedup: 12x
Total time speedup: 3x

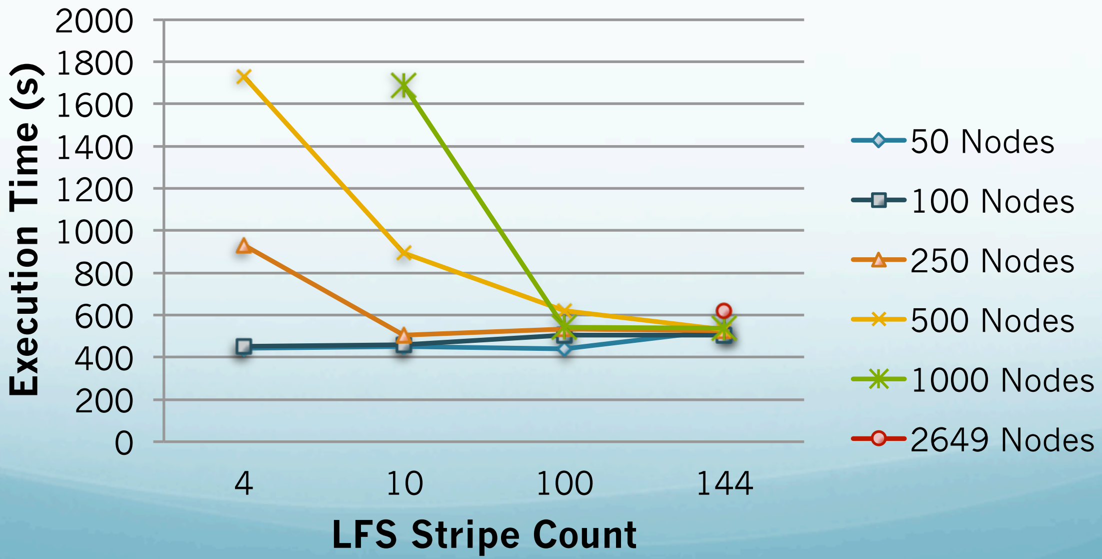
Burst Buffer Science Cases

Seismic Natural Migration



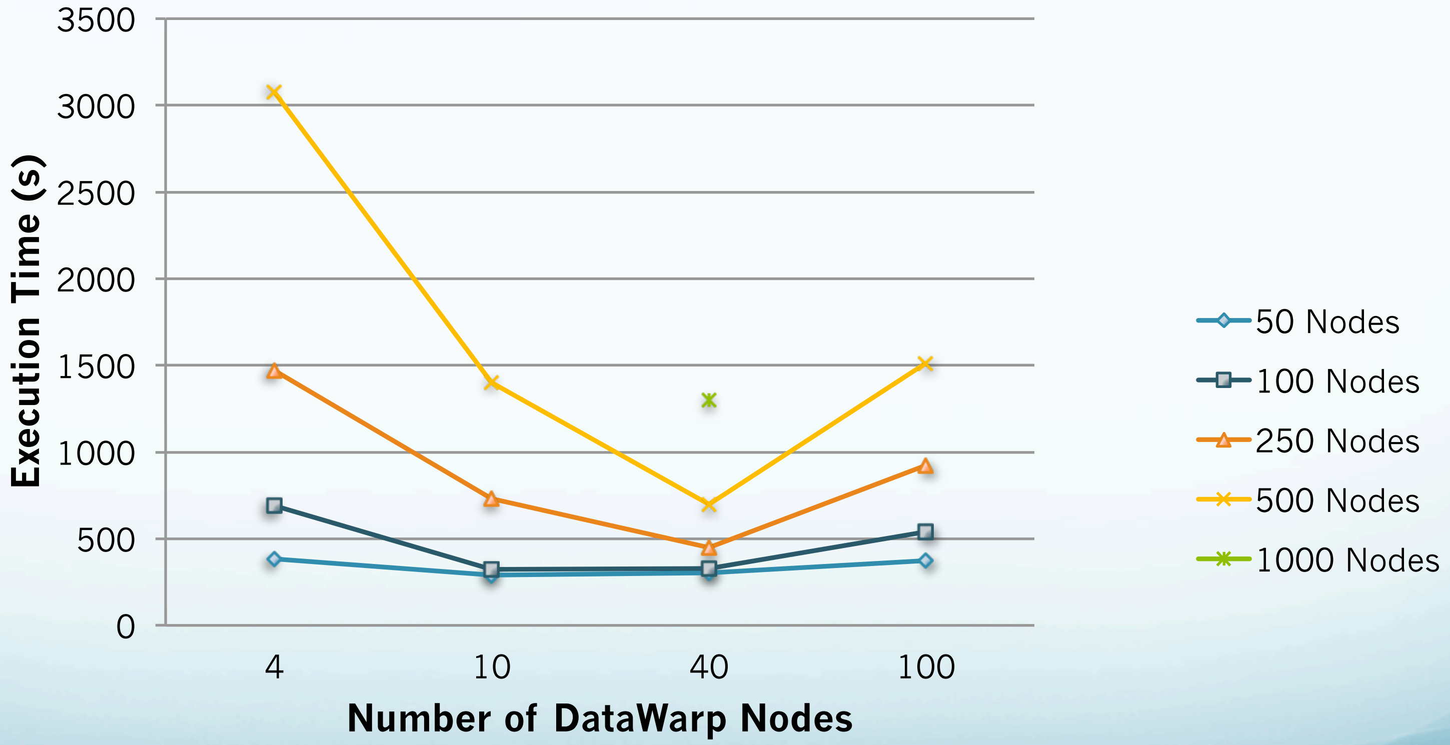


- Natural Migration is a seismic imaging tool that maps buried faults
- The Algorithm uses recorded Green's functions stored in a single file with more than 86GB of size.
- Significant performance improvement by tuning the Lustre stripe count



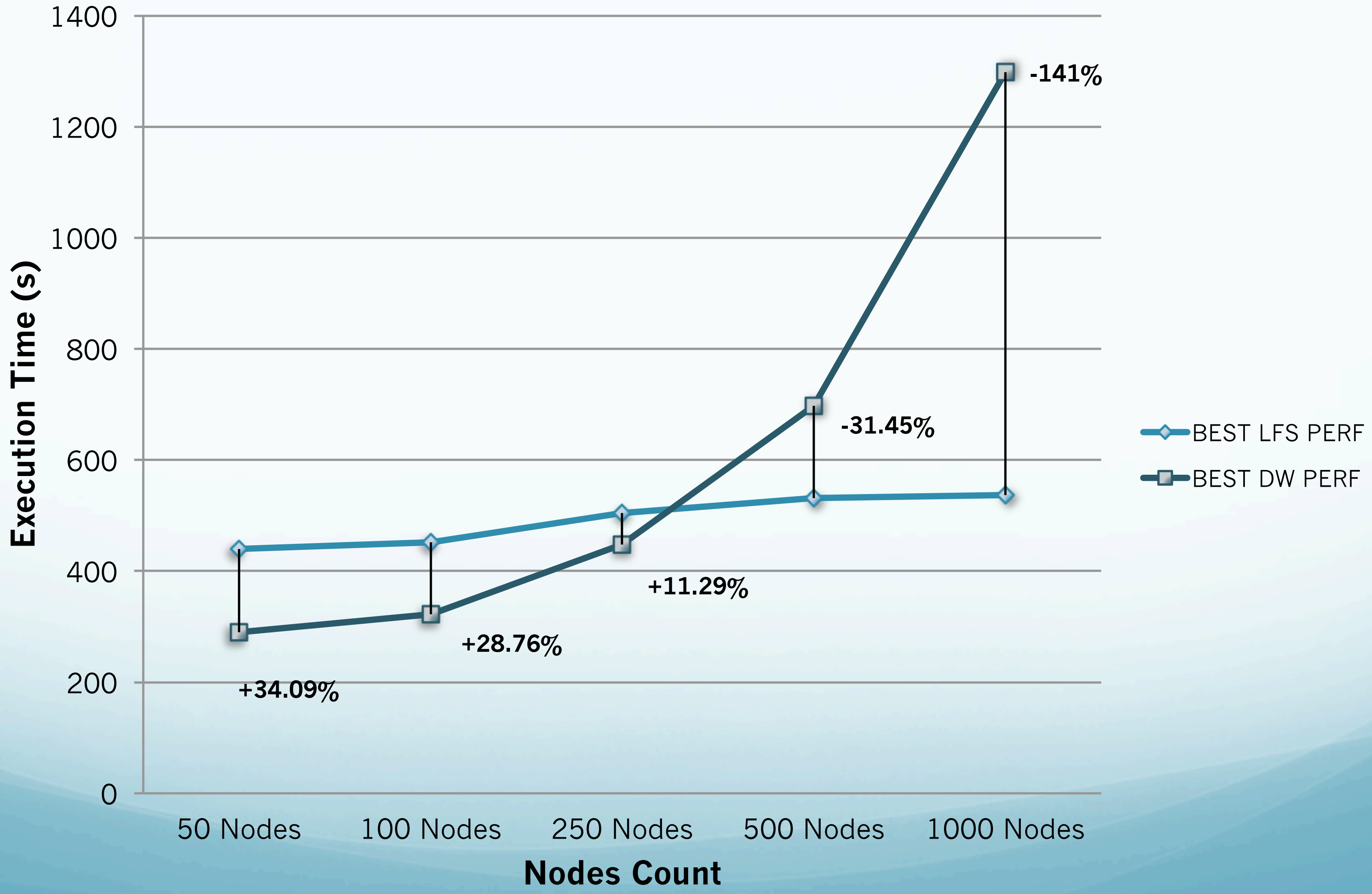
Seismic Natural Migration

- **DataWarp Nodes Count**



Seismic Natural Migration

- Lustre Filesystem vs DataWarp

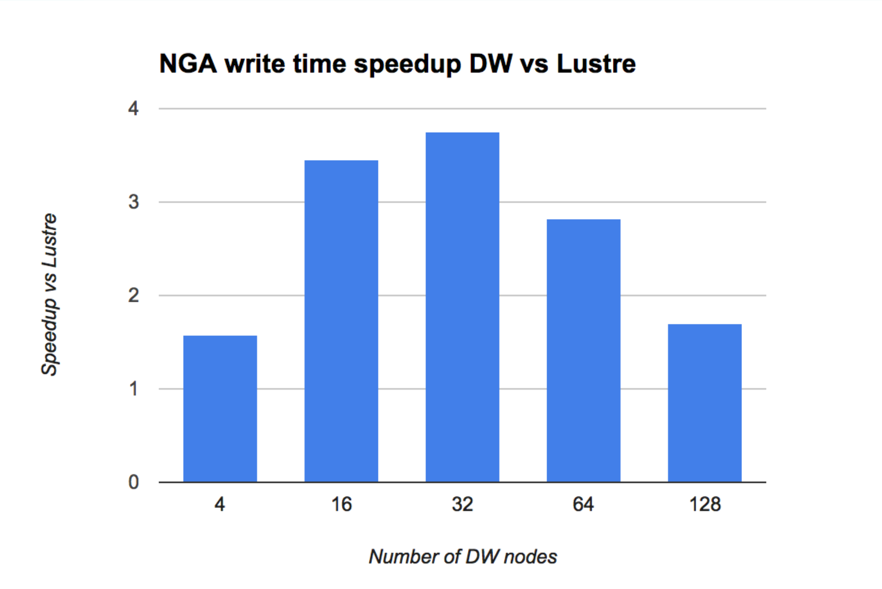


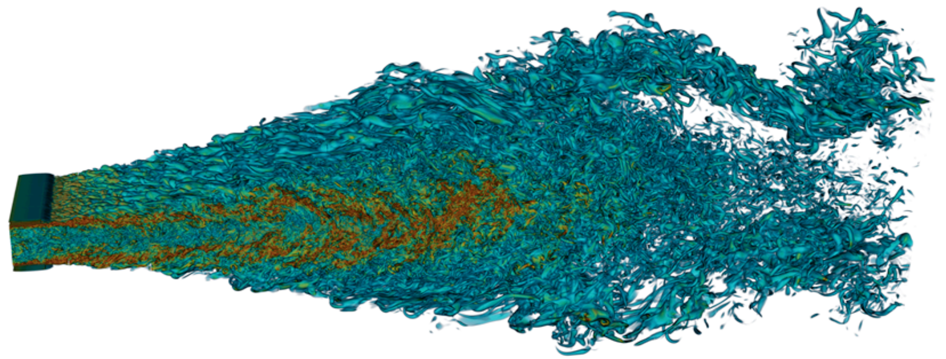
Turbulent partially premixed flames

- NGA is large-eddy simulation (LES) and direct numerical simulation (DNS) code capable of solving the low-Mach number variable density Navier-Stokes equations on structured cartesian and cylindrical meshes.

- NGA scales well across Shaheen2. Productions runs typically use 1024 nodes and writing 590 GB files.

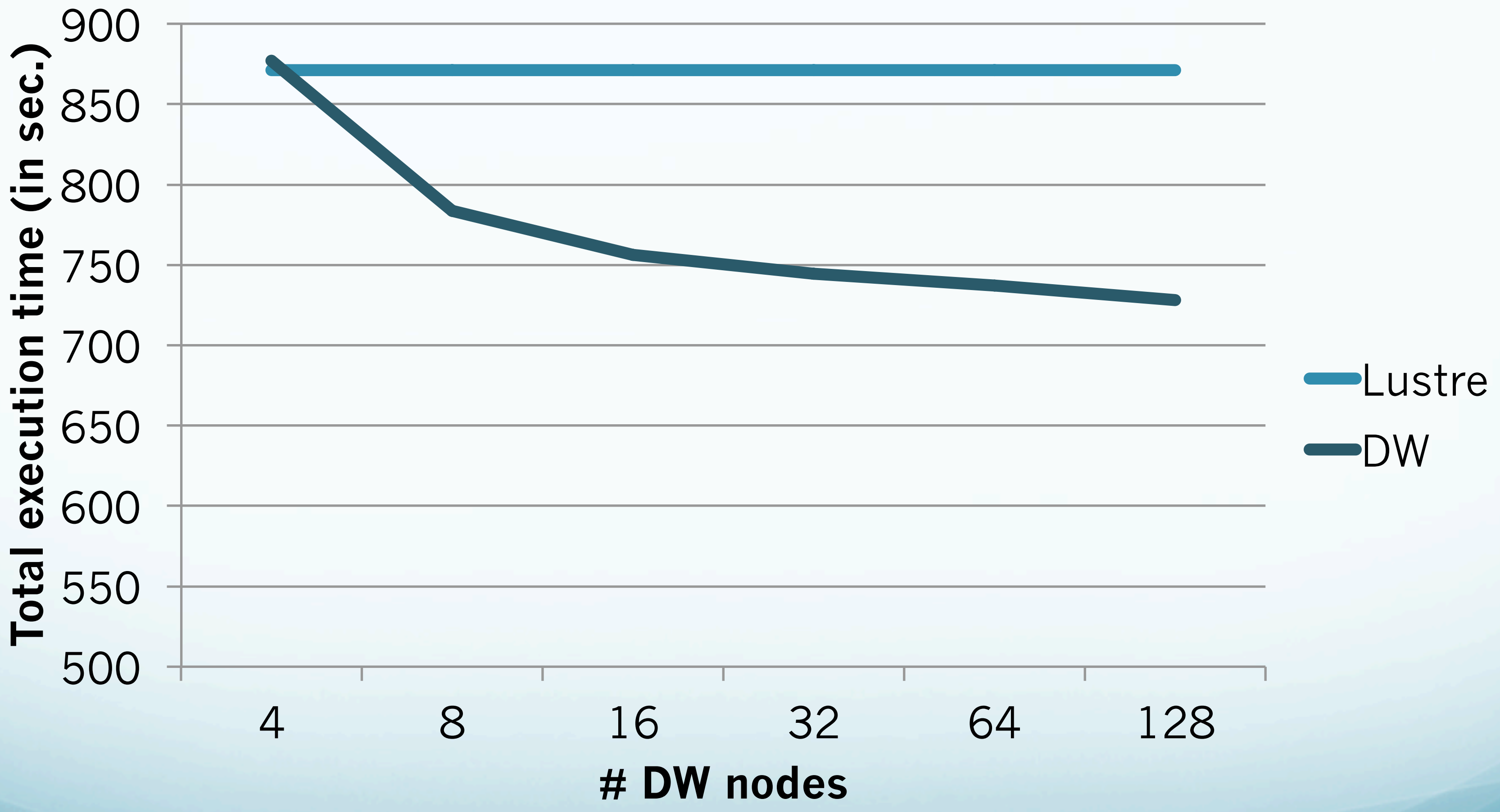
- Burst Buffer demonstrated with up to 3.75 times performance improvement when compared to Lustre (whole simulation 1.24x faster)





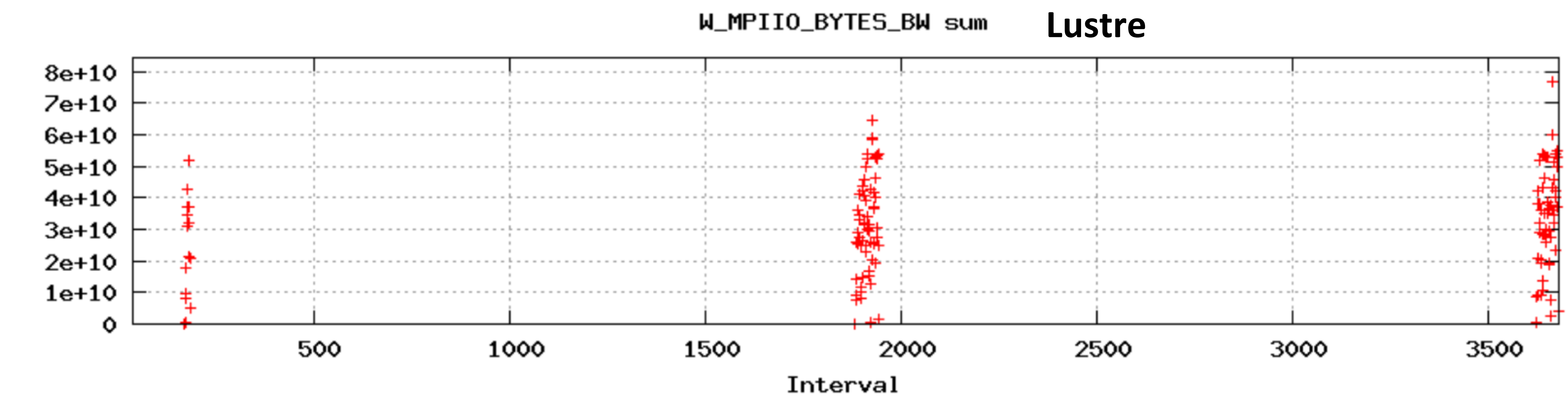
Vorticity magnitude of 22 billion-point methane-air premixed turbulent flame direct numerical simulation (DNS)

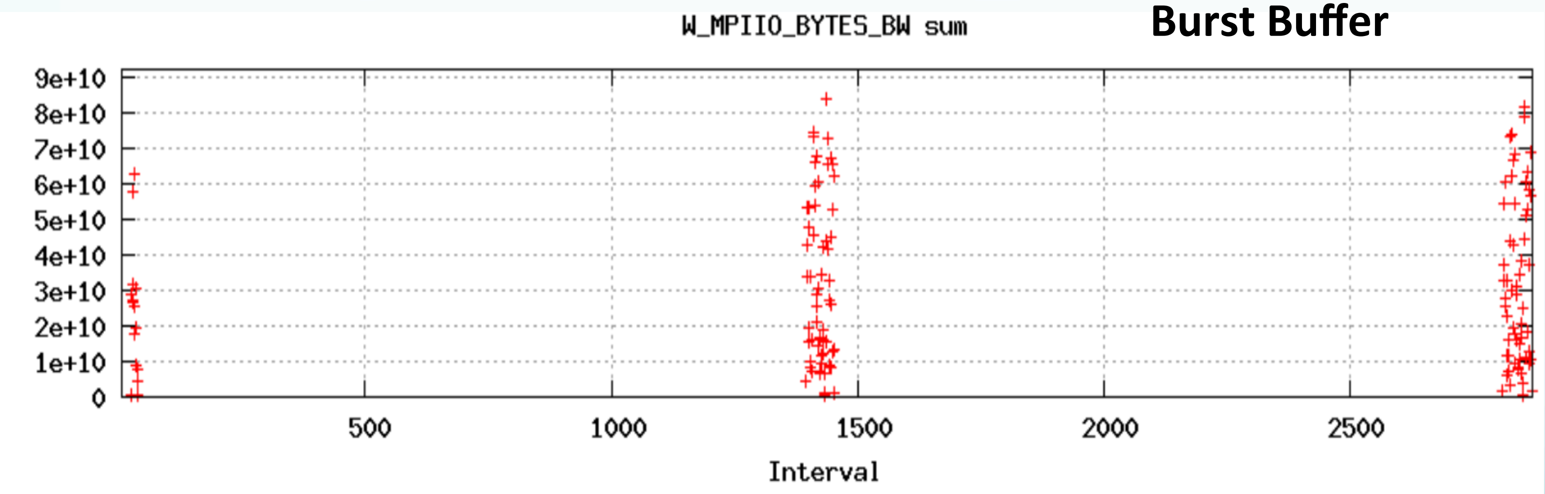
WRF – Lustre vs DW



256 compute nodes 4MPI x 8 OMP per node with PnetCDF

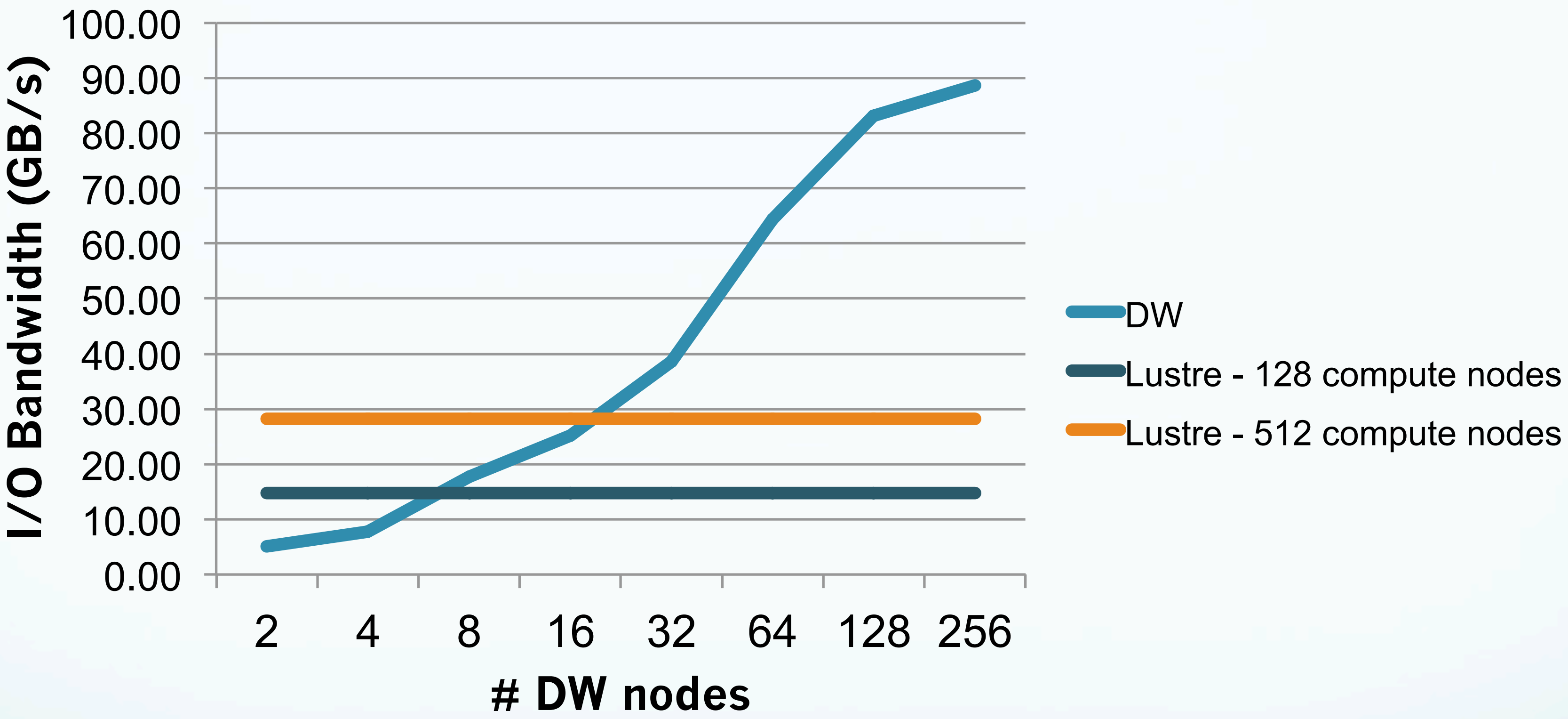
WRF I/O with MPI statistics Lustre vs BB


Lustre


Burst Buffer

Peak performance of DW is above Lustre, Execution faster on DW.

NAS – BT I/O Benchmark - PNetCDF

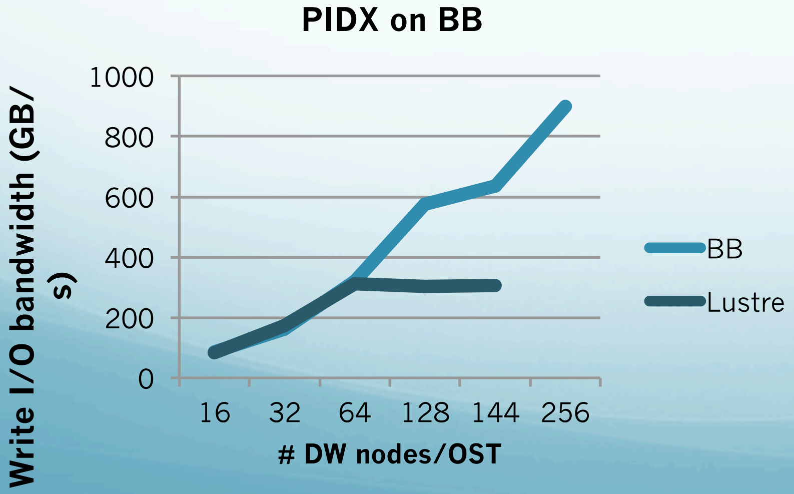


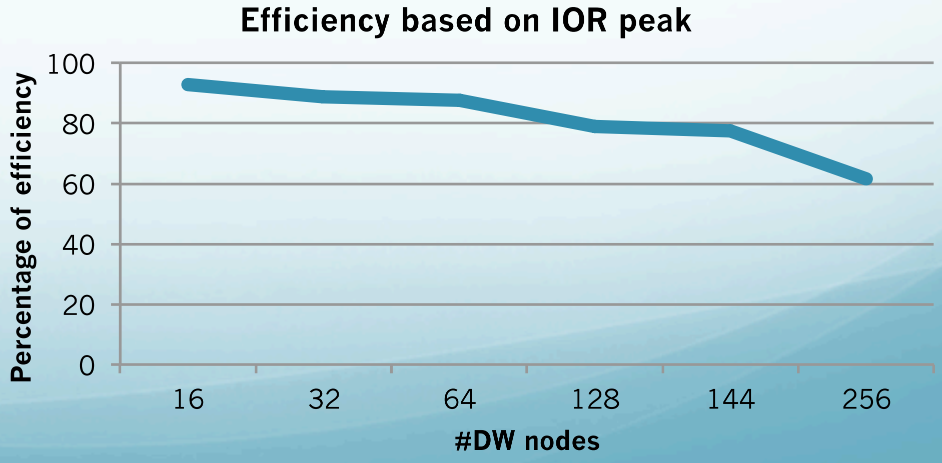
- For DW experiments: 128 compute nodes (32 MPI processes per node)
  - The problem size changes across different number of DW nodes to stress the SSDs.
  - For 2 DW nodes we save 400GB PNetCDF file, for 256 DW nodes, we save 51TB file

- For Lustre (144 OSTs)
  - 128 compute nodes which is the maximum that DW runs used
  - Used also 512 compute nodes (16384 MPI processes) to help Lustre but still falls behind.

PIDX

- PIDX is an efficient parallel I/O library that reads and writes multiresolution IDX data files, providing high scalability up to 768k cores. Optimized data movement with 3 phase I/O (restructuring, aggregation, I/O write)
- Successful integration with several simulation codes
  - KARFS (KAUST Adaptive Reacting Flow Solvers) on Shaheen II, Uintah (Mira), S3D
- Performance 32 cores per node and we save 64 MB per core (realistic case)





Conclusions

- Burst Buffer is easy to use and significant improvement are obtained by editing the job script thanks to to the workload manager SLURM.
- Burst Buffer is still at early development, more features are coming.
- Need to select the best number of DW nodes for best performance
  - Scaling study is needed , otherwise Lustre might be faster.
- More effort on tuning I/O strategies is needed to achieve better efficiency.

THANKS