Integrating Darshan I/O Performance Monitoring Into SPOT

Author: Wesley Kwiecinski

Mentors: Rui Wang (ANL), Peter Van Gemmeren (ANL)

Advisor: Michael E. Papka (UIC / ANL)

HEP-CCE/SOP

Overview

- Brief recap
- Summer work
- Integration into Software Performance Optimization Team (SPOT) Performance Monitoring Board (PMB)
- Future Work

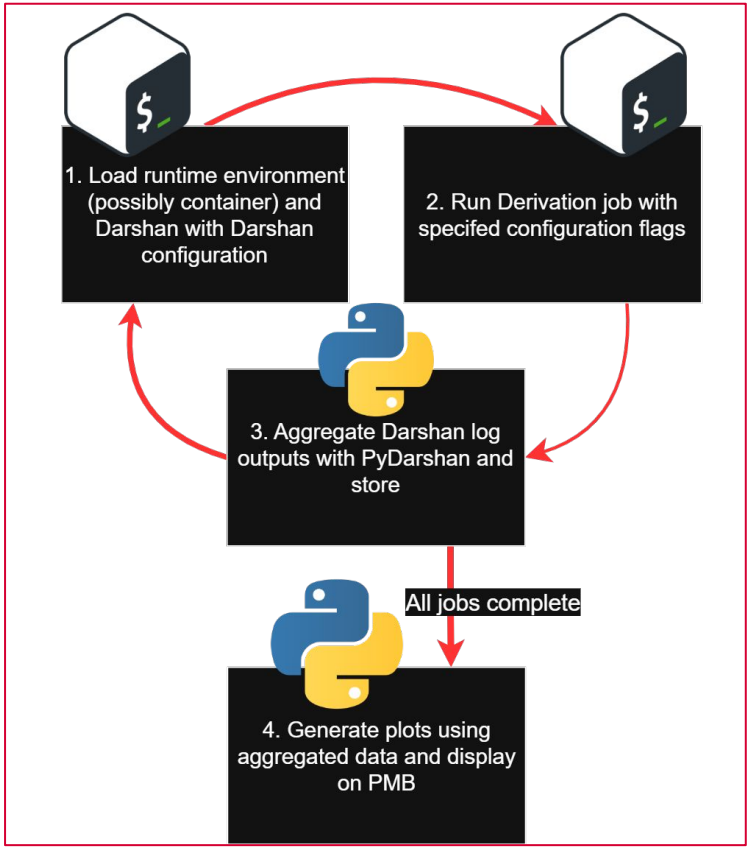
Brief Recap...

- Darshan would extend SPOT's tools with extra information for analysis
  - File timestamps, categories of I/O, modules for various I/O contexts (POSIX, STDIO, HDF5, etc)
  - Trace hotspots between Athena releases w/ timestamps (DXT)
  - Darshan extends SPOT with support for tracking individual workers
  - Darshan traces files per process
- Darshan has negligible overhead on HEP workflows*

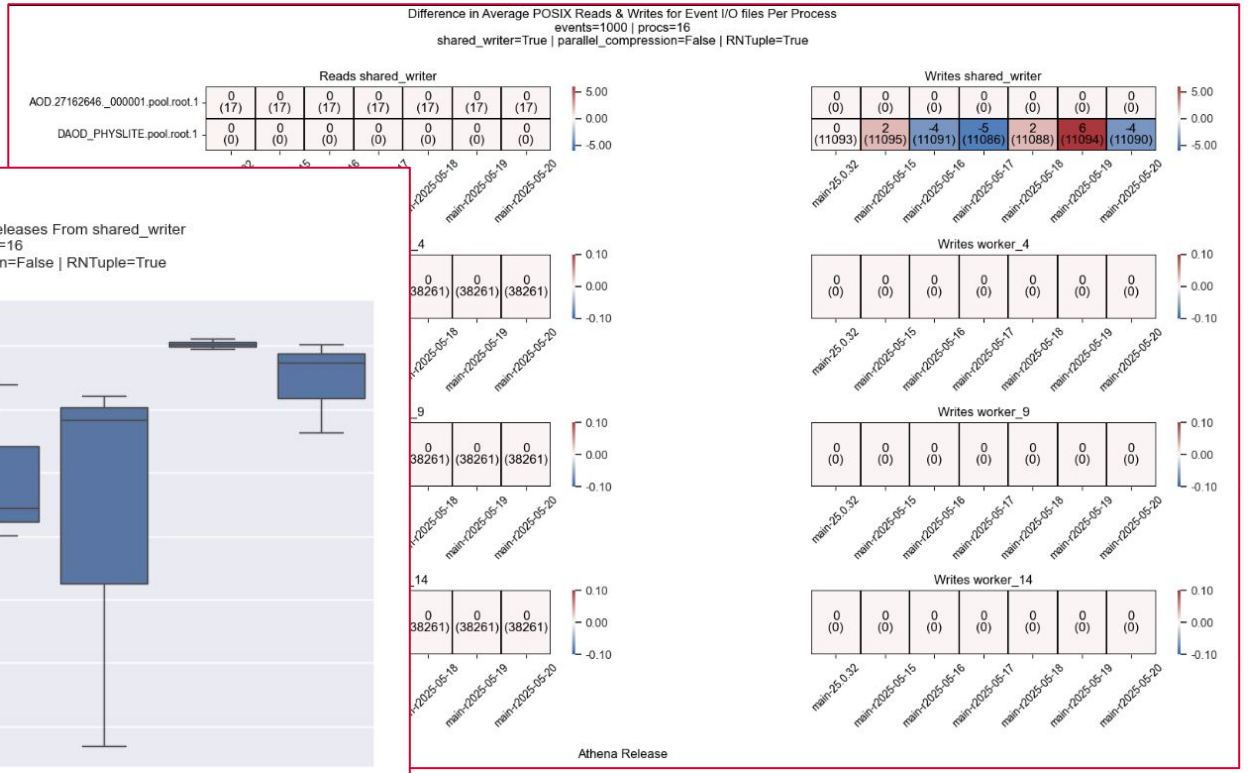
* A. Vijayakumar. (2023). I/O Monitoring for portable HPC applications

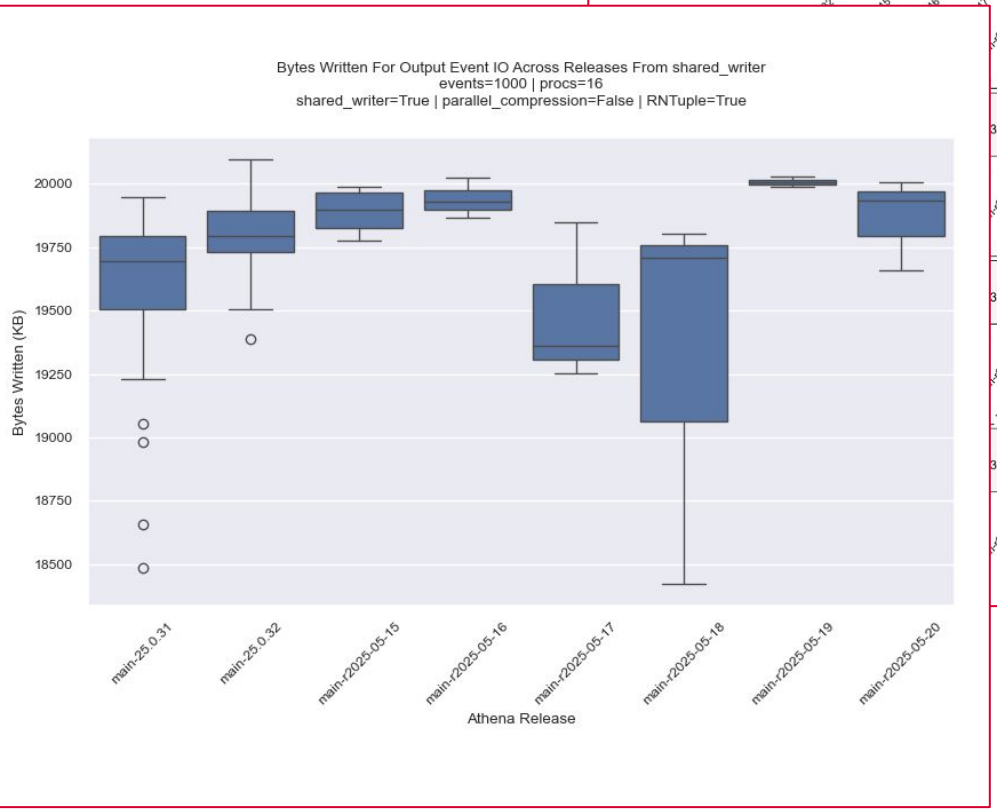
Brief Recap...

- Use Cron + container + configurable bash script
  - Track I/O performance of Derivations w/ different configurations and releases



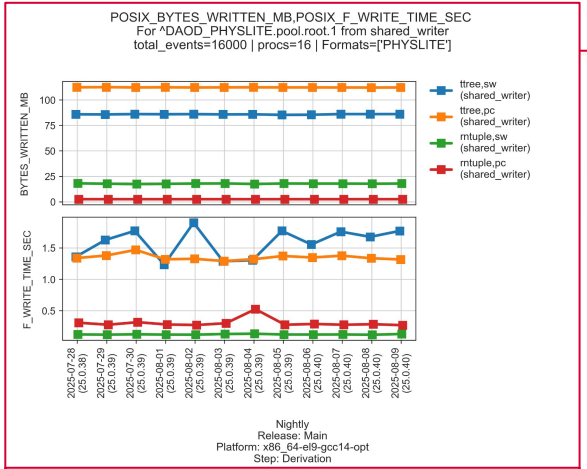
Brief Recap...

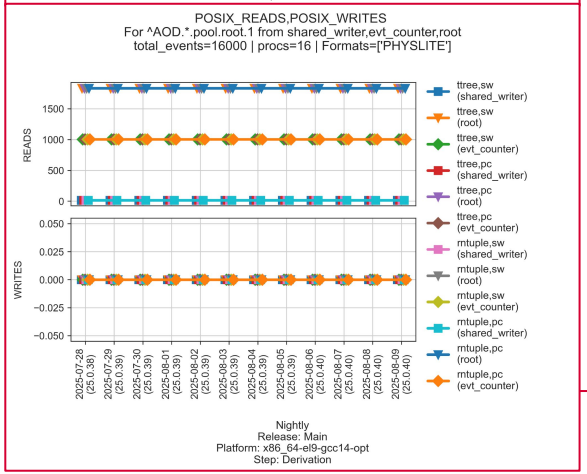

Difference in Average POSIX Reads & Writes for Event I/O files Per Process
events=1000 | procs=16
shared_writer=True | parallel_compression=False | RNTuple=True
Reads shared_writer
Writes shared_writer
Writes worker_4
Writes worker_9
Writes worker_14
AOD.27162646._000001.pool.root.1
DAOD_PHYSLITE.pool.root.1
main-25.0.32
main-r2025-05-15
main-r2025-05-16
main-r2025-05-17
main-r2025-05-18
main-r2025-05-19
main-r2025-05-20
Athena Release


Bytes Written For Output Event IO Across Releases From shared_writer
events=1000 | procs=16
shared_writer=True | parallel_compression=False | RNTuple=True
Bytes Written (KB)
20000
19750
19500
19250
19000
18750
18500
main-25.0.31
main-25.0.32
main-r2025-05-15
main-r2025-05-16
main-r2025-05-17
main-r2025-05-18
main-r2025-05-19
main-r2025-05-20
Athena Release

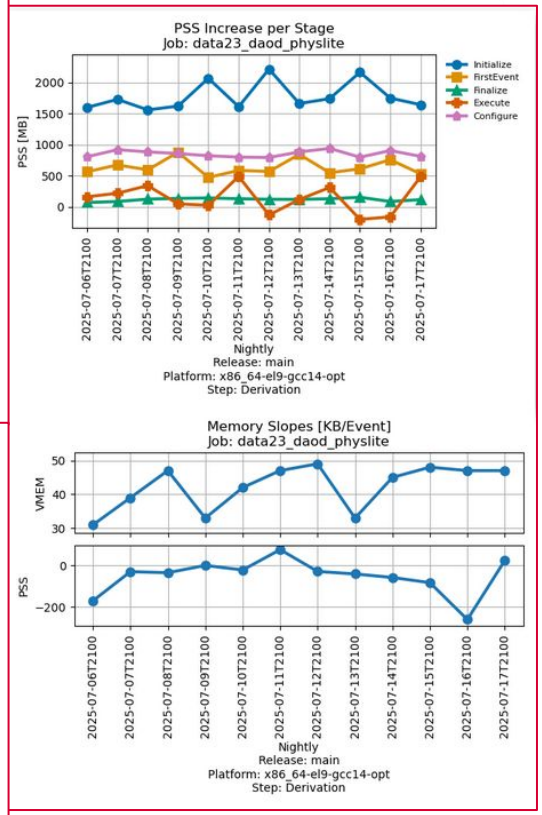
Summer Work

- Homogenization of plots to match SPOT PMB
- Internal work on plot generation scripts
- Gathering DXT data for more details on event I/O reads & writes
- Leaning into gathering data from 4 different Derivation configurations
- Integration with SPOT's PMB



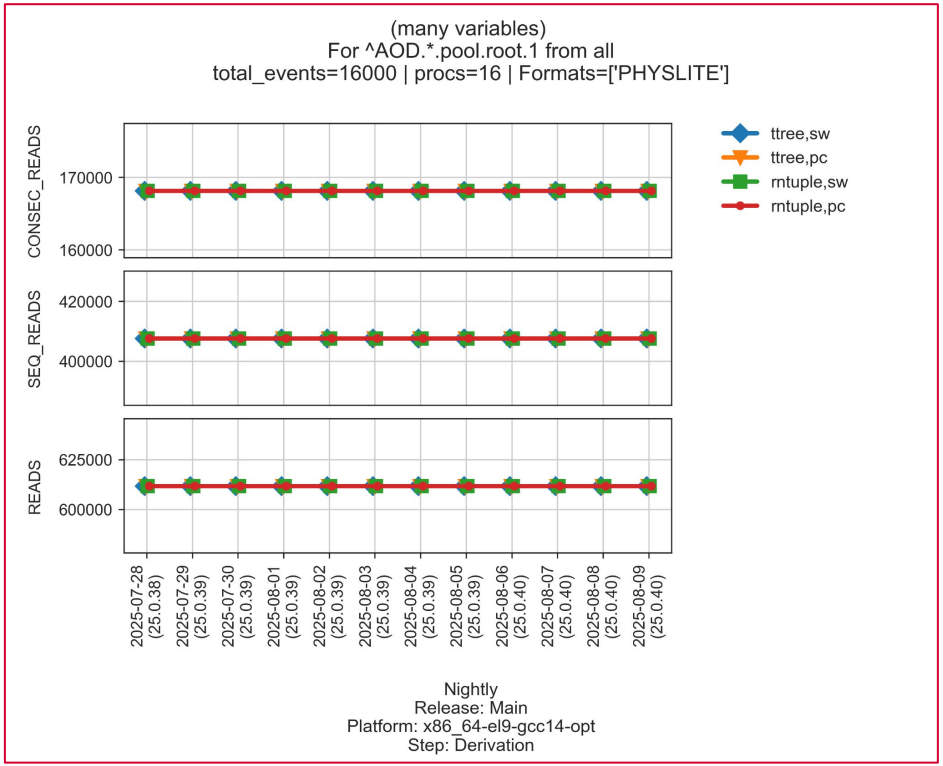


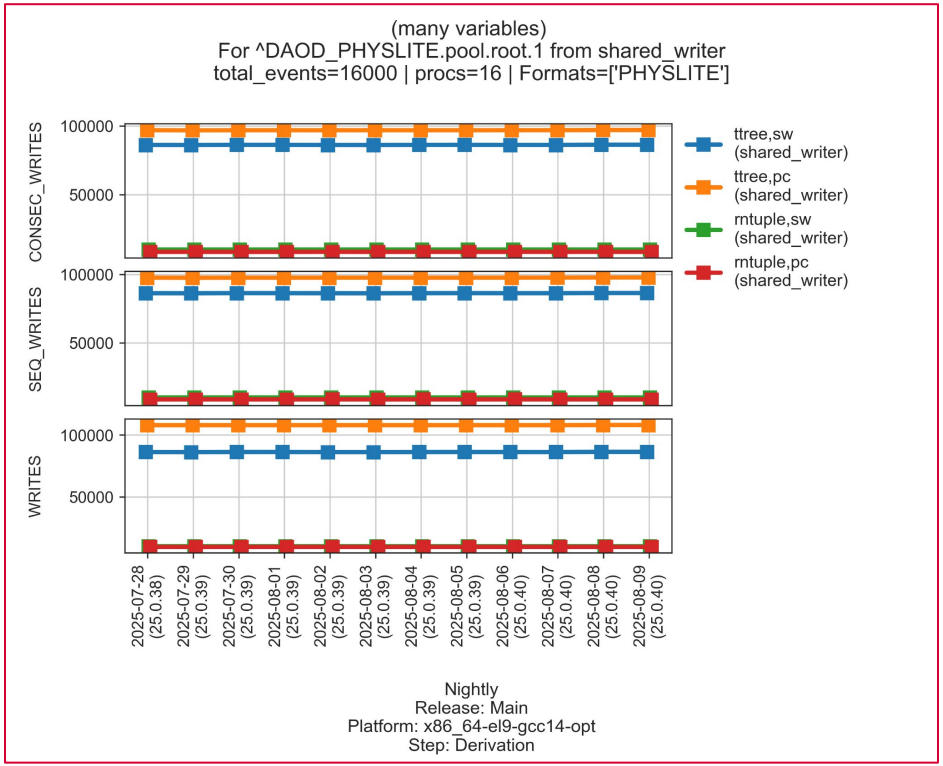
New data plots



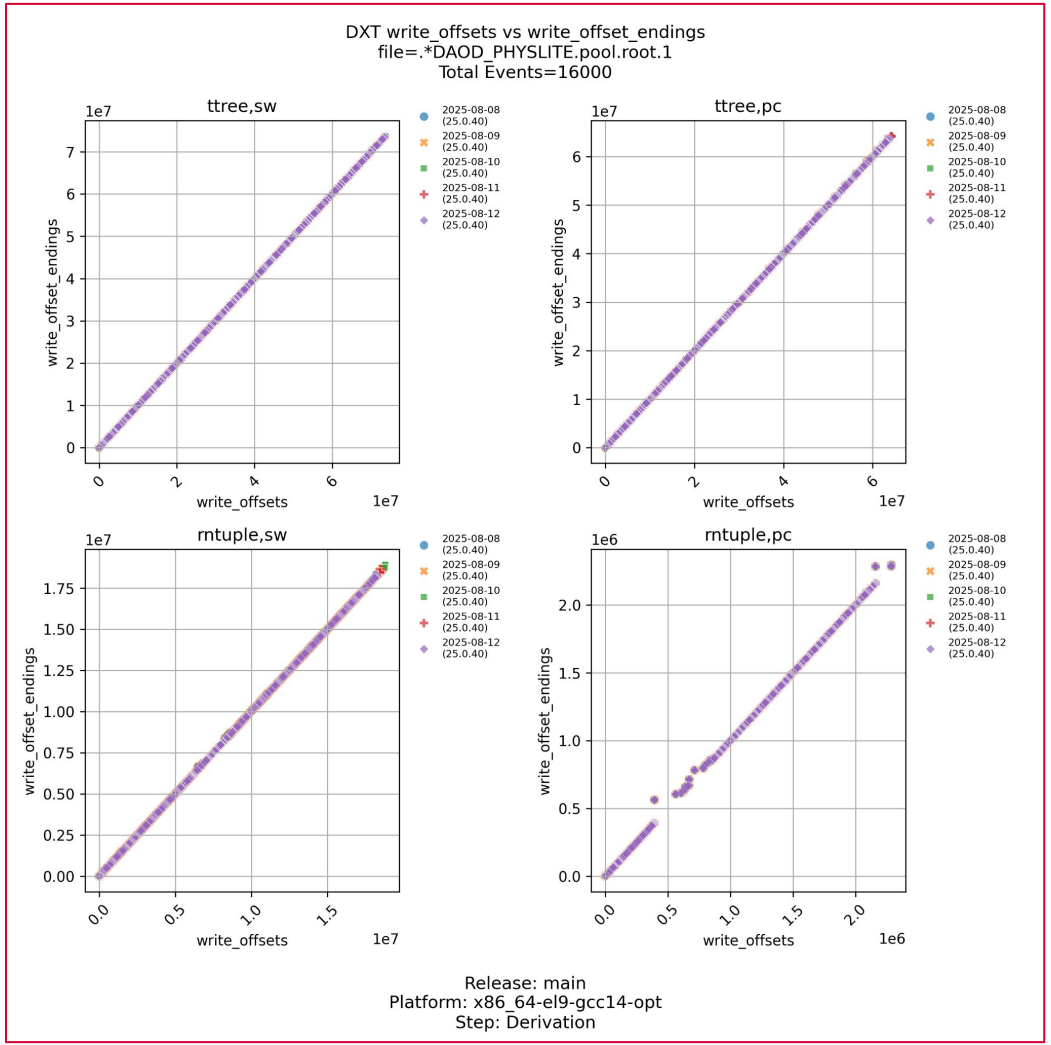
PMB

Summer Work

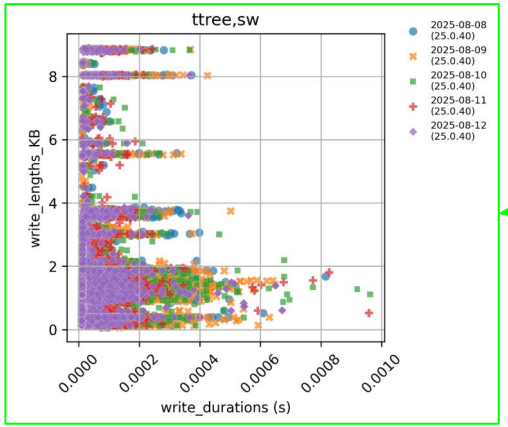

(many variables)
For ^AOD.*.pool.root.1 from all
total_events=16000 | procs=16 | Formats=['PHYSLITE']
CONSEC_READS
SEQ_READS
READS
ttree,sw
ttree,pc
rntuple,sw
rntuple,pc
2025-07-28 (25.0.38)
2025-07-29 (25.0.39)
2025-07-30 (25.0.39)
2025-08-01 (25.0.39)
2025-08-02 (25.0.39)
2025-08-03 (25.0.39)
2025-08-04 (25.0.39)
2025-08-05 (25.0.39)
2025-08-06 (25.0.40)
2025-08-07 (25.0.40)
2025-08-08 (25.0.40)
2025-08-09 (25.0.40)
Nightly
Release: Main
Platform: x86_64-el9-gcc14-opt
Step: Derivation

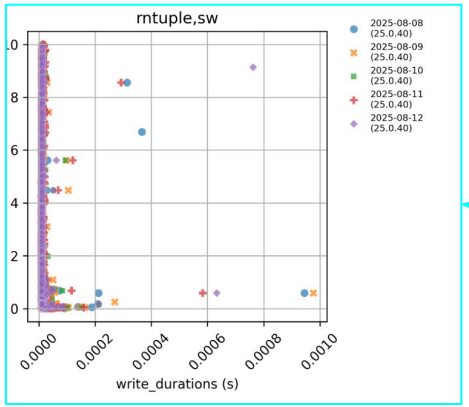

(many variables)
For ^DAOD_PHYSLITE.pool.root.1 from shared_writer
total_events=16000 | procs=16 | Formats=['PHYSLITE']
CONSEC_WRITES
SEQ_WRITES
WRITES
ttree,sw (shared_writer)
ttree,pc (shared_writer)
rntuple,sw (shared_writer)
rntuple,pc (shared_writer)
2025-07-28 (25.0.38)
2025-07-29 (25.0.39)
2025-07-30 (25.0.39)
2025-08-01 (25.0.39)
2025-08-02 (25.0.39)
2025-08-03 (25.0.39)
2025-08-04 (25.0.39)
2025-08-05 (25.0.39)
2025-08-06 (25.0.40)
2025-08-07 (25.0.40)
2025-08-08 (25.0.40)
2025-08-09 (25.0.40)
Nightly
Release: Main
Platform: x86_64-el9-gcc14-opt
Step: Derivation

Summer Work

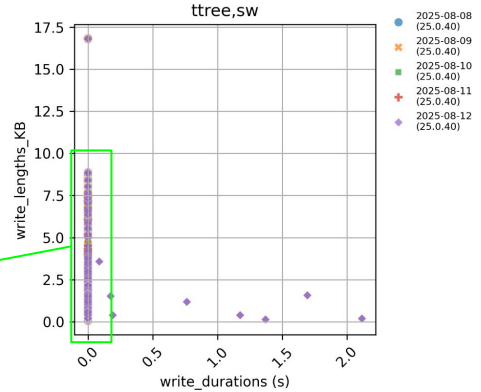

DXT write_offsets vs write_offset_endings
file=.*DAOD_PHYSLITE.pool.root.1
Total Events=16000
ttree,sw
ttree,pc
rntuple,sw
rntuple,pc
write_offset_endings
write_offsets
1e7
1e6
2025-08-08 (25.0.40)
2025-08-09 (25.0.40)
2025-08-10 (25.0.40)
2025-08-11 (25.0.40)
2025-08-12 (25.0.40)
Release: main
Platform: x86_64-el9-gcc14-opt
Step: Derivation

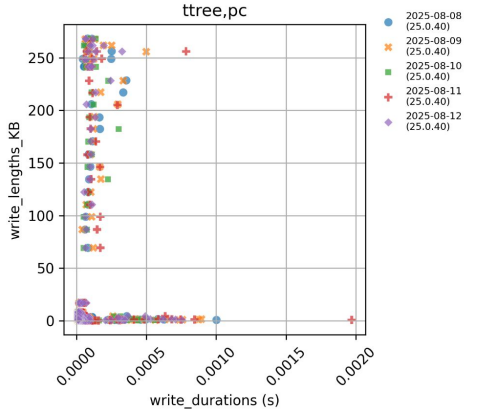
Summer Work

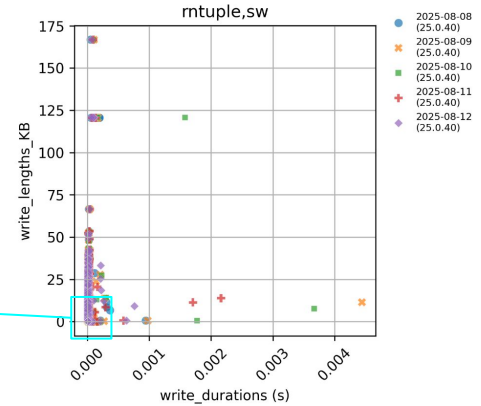


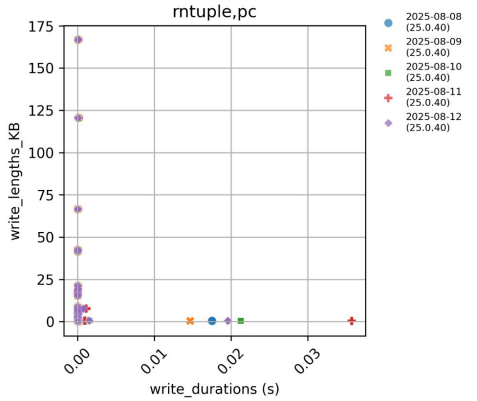


DXT write_durations (s) vs write_lengths_KB
file=DAOD_PHYSLITE.pool.root.1
Total Events=16000



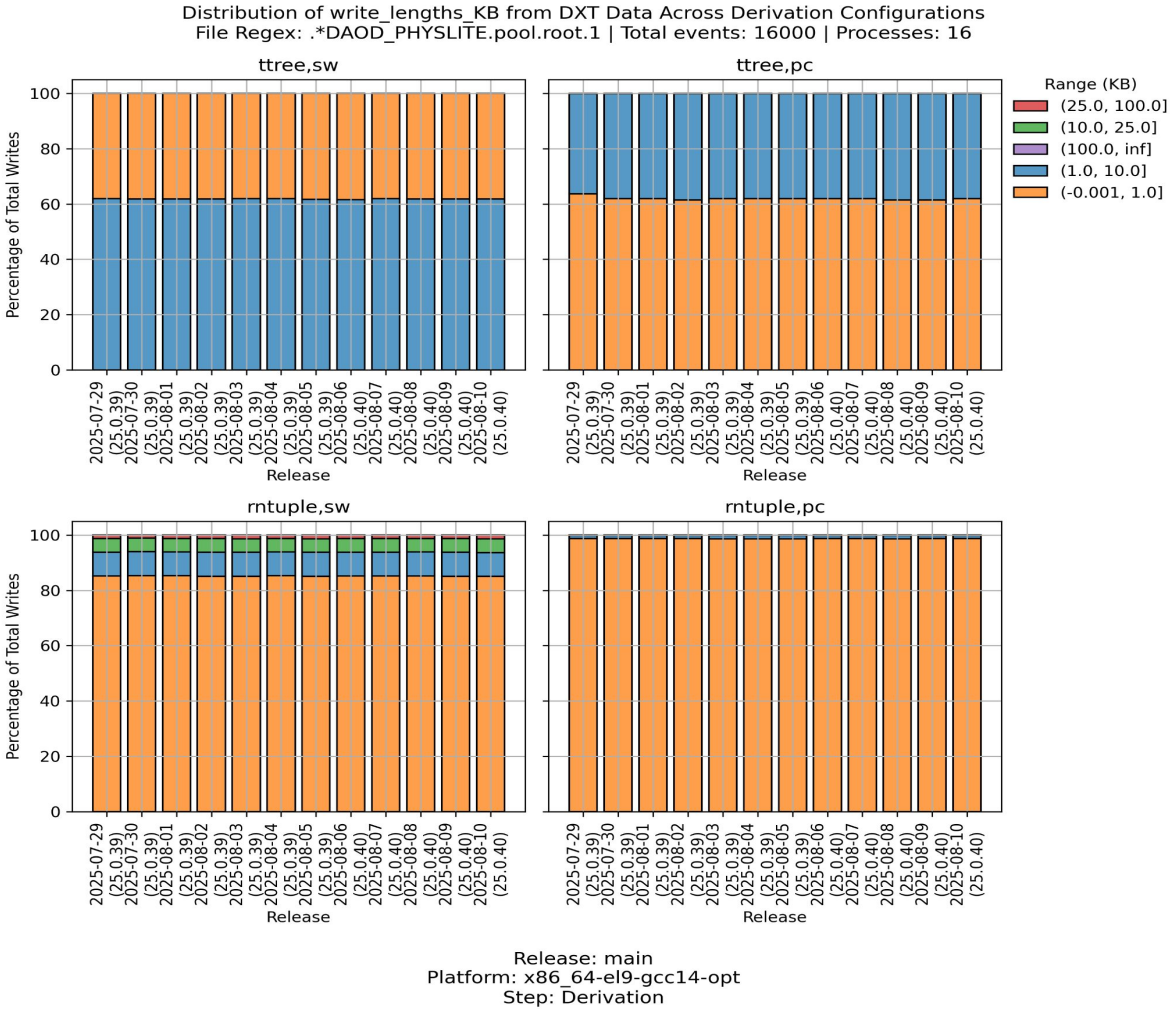




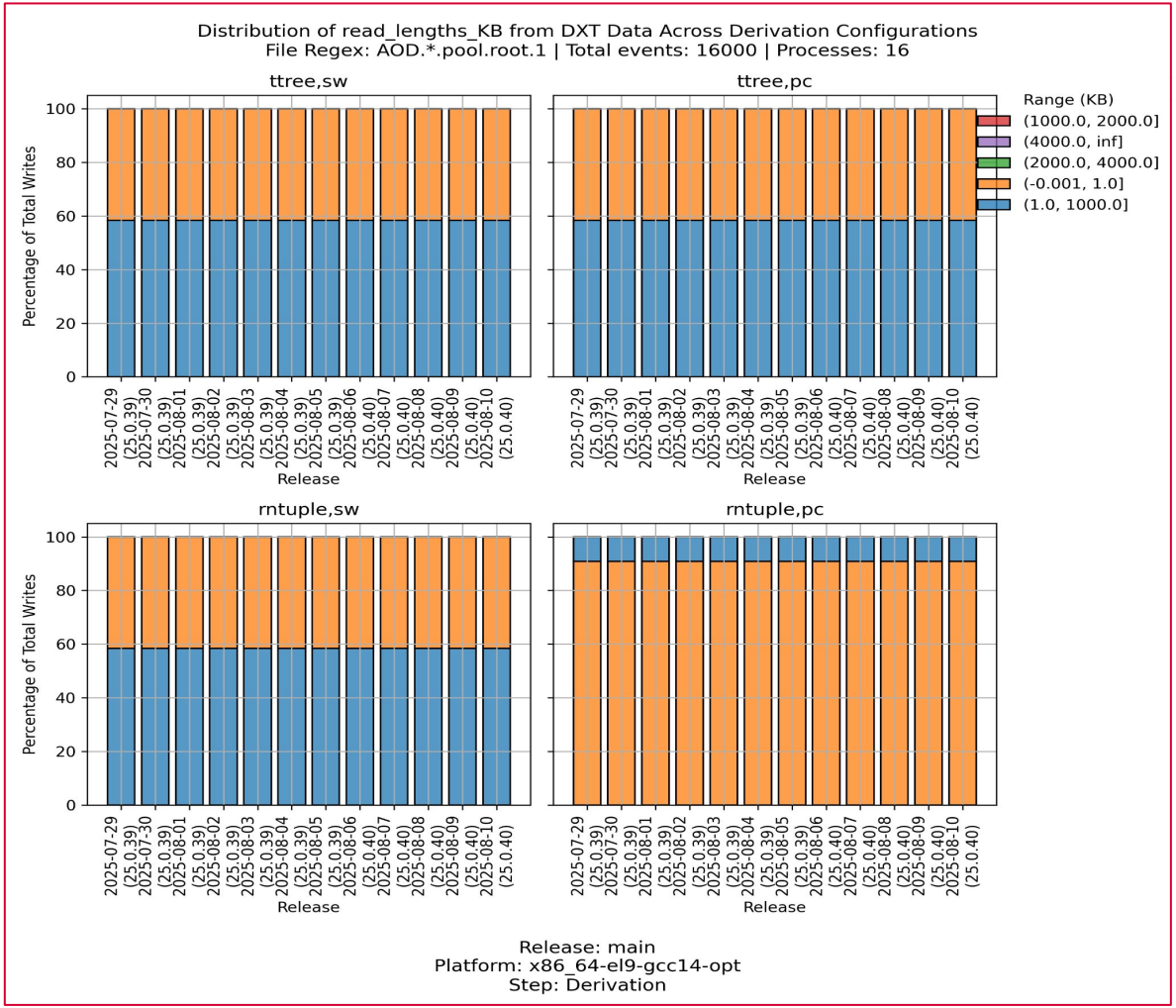


Release: main
Platform: x86_64-el9-gcc14-opt
Step: Derivation

Summer Work


Distribution of write_lengths_KB from DXT Data Across Derivation Configurations
File Regex: .*DAOD_PHYSLITE.pool.root.1 | Total events: 16000 | Processes: 16
ttree,sw
ttree,pc
rntuple,sw
rntuple,pc
Percentage of Total Writes
Release
Range (KB)
(25.0, 100.0]
(10.0, 25.0]
(100.0, inf]
(1.0, 10.0]
(-0.001, 1.0]
Release: main
Platform: x86_64-el9-gcc14-opt
Step: Derivation

Summer Work


Distribution of read_lengths_KB from DXT Data Across Derivation Configurations
File Regex: AOD.*.pool.root.1 | Total events: 16000 | Processes: 16
ttree,sw
ttree,pc
rntuple,sw
rntuple,pc
Percentage of Total Writes
Release
Range (KB)
(1000.0, 2000.0]
(4000.0, inf]
(2000.0, 4000.0]
(-0.001, 1.0]
(1.0, 1000.0]
Release: main
Platform: x86_64-el9-gcc14-opt
Step: Derivation

Integration of Scripts with SPOT PMB

- Spent time over the summer migrating job & plotting scripts to SPOT's PMB repository
- Merged with SPOTs PMB this week, starting to collect data
- Still need to include DXT plots in PMB, currently only using plots from posix module

Future Work

- Adjust DXT data collection
- Collect data from SPOT servers to identify possible I/O bottlenecks with Derivations
  - Still need to include DXT data, PMB will only show standard POSIX data for now
- Potential Master's Thesis topic
  - Currently exploring performance of small I/O in Derivations on machines with low vs high I/O contention
  - I/O Performance & I/O Contention on HPCs are generally well-studied, some case studies on different scientific software

Thank You!

Acknowledgements

- This work is a collaboration between the University of Illinois Chicago and Argonne National Laboratory under the C2-The-P2 fellowship
- This work uses resources at the Argonne Laboratory Computing Resource Center (LCRC)