DARSHAN FOR ATHENA

Doug Benjamin[1], Peter Van Gemmeren[2], Shane Snyder[2], ***Rui Wang***[2]
1. Brookhaven National Laboratory
2. Argonne National Laboratory

HEP-CCE All-Hands Meeting
Oct. 11, 2022

Athena I/O monitoring

- Using Darshan as the I/O monitoring tool for Atlas HPC workflow
- Gain deeper insights into I/O patterns of Athena

- Load Darshan library directly in athena & exclude /cvmfs activities in runtime environment

```
> head log.EVNTtoHITS
11:00:45 Thu Oct  6 11:00:45 CDT 2022
11:00:45 Preloading /lcrc/group/ATLAS/users/
rwang/Argonne_computing/PPS-CCE/darshan/
build_darshan/dev-fork-child-issue786/lib/
libdarshan.so
11:00:45 ##########################
11:00:45 ##### DARSHAN CONFIG #####
11:00:45 ##########################
```

```
# enable DXT modules, which are off by default
MOD_ENABLE      DXT_POSIX,DXT_MPIIO

# allocate 4096 file records for POSIX and MPI-IO modules
# (darshan only allocates 1024 per-module by default)
MAX_RECORDS     5000      POSIX

# the '*' specifier can be used to apply settings for all modules
# in this case, we want all modules to ignore record names
# prefixed with "/home" (i.e., stored in our home directory),
# with a superseding inclusion for files with a ".out" suffix)
NAME_EXCLUDE    .pyc$,^/cvmfs,^/lib64,^/lib,^/blues/gpfs/home/software *
NAME_INCLUDE    .pool.root.*   *

# bump up Darshan's default memory usage to 8 MiB
MODMEM  8

# avoid generating logs for git and ls binaries
APP_EXCLUDE     which

# only retain DXT traces for files that were accessed
# using small I/O ops 20+% of the time
DXT_SMALL_IO_TRIGGER    .2
```

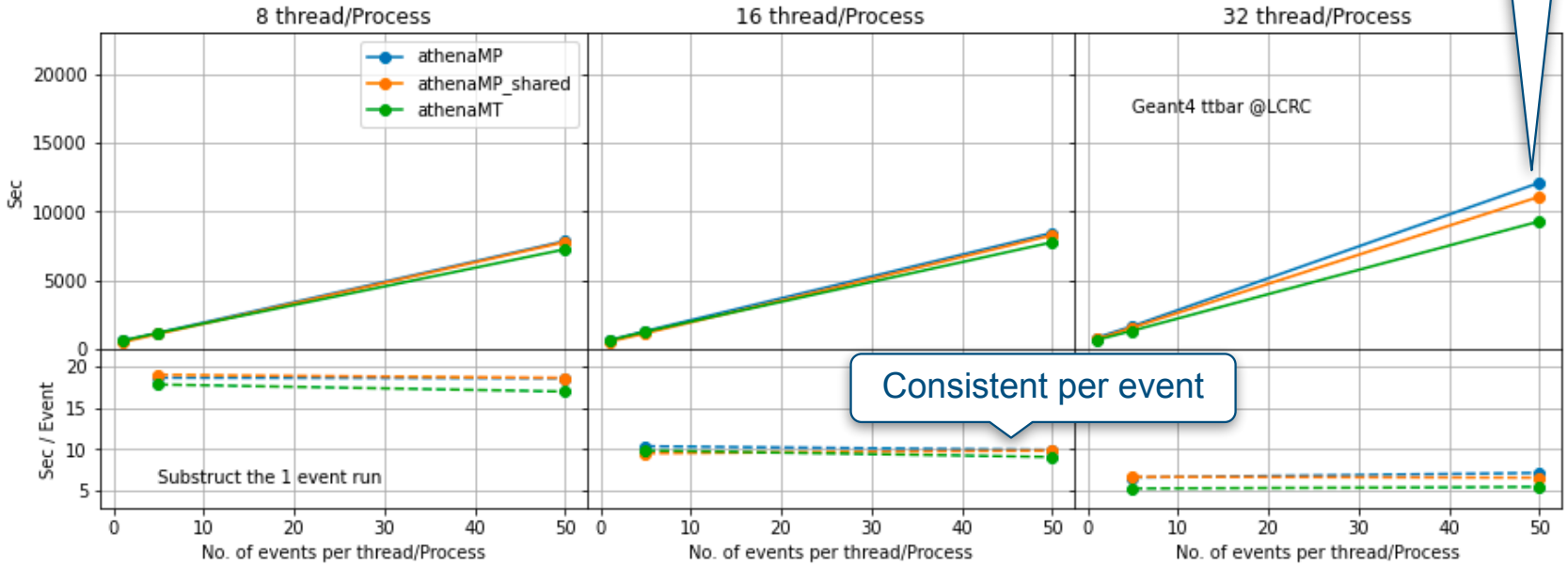
Various Athena Modes

- AthenaMP — multi-Process
  - Independent parallel workers are forked from main process with shared memory allocation
  - Each worker produces its own outputs and merged later via a post-processing merge process
- AthenaMP+SharedWriter — multi-Process
  - A shared writer process does all the output writes
- AthenaMT — multi-thread
  - Gaudi task scheduler maps task to kernel threads
  - Shared single pool of heap memory

Wall time

- Total running time of the jobs: Simulation (+ Merging)

As expected
- AthenaMP takes a bit more time due to additional merging proc
- AthenaMT is slightly fast than athenaMPs



Note: Merging is relatively fast for Simulation, as Simulation is very CPU intense. For ATLAS Derivation, which is I/O dominated, merging had become bottleneck and SharedWriter was&is deployed in Run2 and 3.

Darshan records

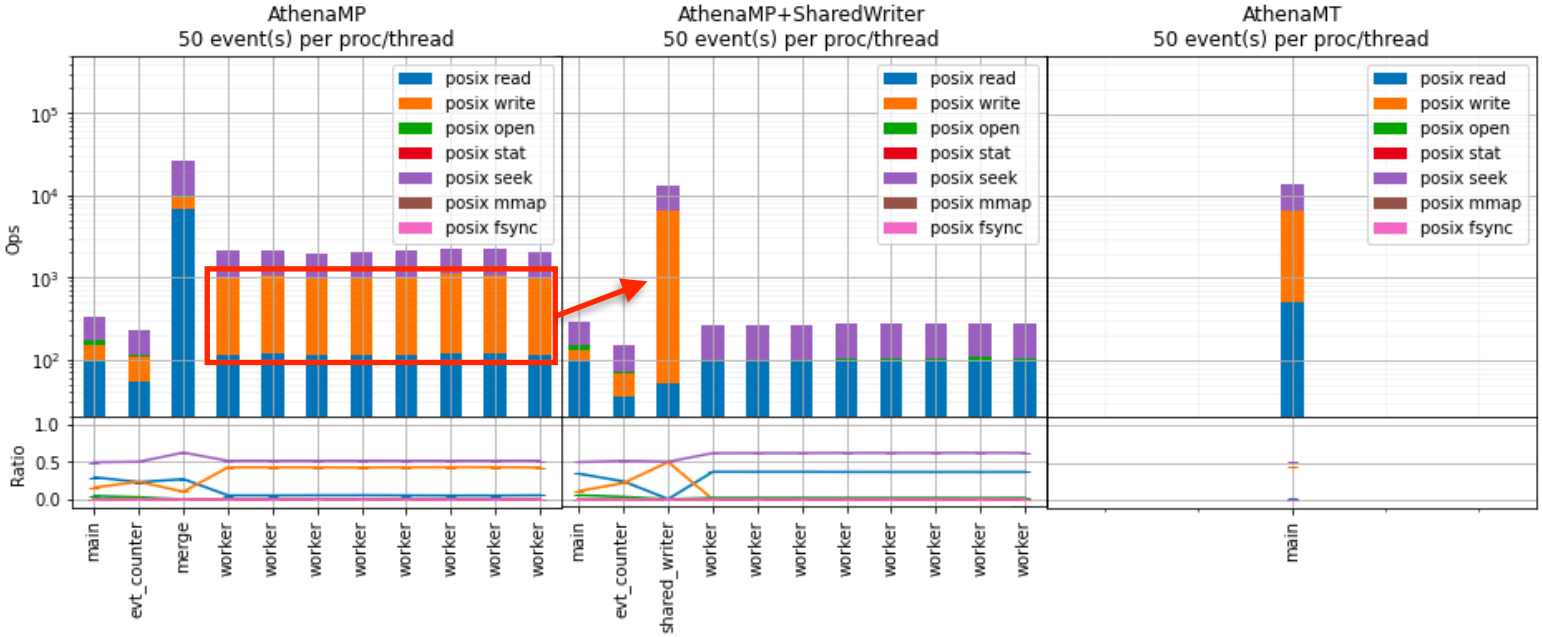
- Only looking at the I/O behavior of the input&output event data files
- Filter out all other records

{'runargs.EVNTtoHITS.py', 'eventLoopHeartBeat.txt', 'oflcond.000001.conditions.recon.pool.v0000._0058.pool.root', 'SimParams.db-journal', 'cond12_data.000029.gen.COND._0001.pool.root', '<STDERR>', '<STDOUT>', 'cond09_mc.000118.gen.COND._0003.pool.root', 'heatmap:POSIX', 'cond09_mc.000114.gen.COND._0003.pool.root', 'perfmonmt.json', 'test.HITS.pool.root', 'cond08_mc.000003.gen.COND._0064.pool.root', 'cond09_mc.000113.gen.COND._0001.pool.root', '<STDIN>', 'cond09_mc.000019.gen.COND._0010.pool.root', 'PoolFileCatalog.xml', 'athenaMT_FullG4_ttbar_8_1', 'hostnamelookup.tmp', 'cond09_mc.000010.gen.COND._0002.pool.root', 'libdarshan.so', 'valid1.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.evgen.EVNT.e4993.EVNT.08166201._000012.pool.root.1', 'SimParams.db', 'oflcond.000002.conditions.simul.pool.v0000._0029.pool.root__DQ2-1250194490', 'heatmap:STDIO'}

{'test.HITS.pool.root', ← **output file**
'valid1.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.evgen.EVNT.e4993.EVNT.08166201._000012.pool.root.1'} ← **input file**

POSIX Operations

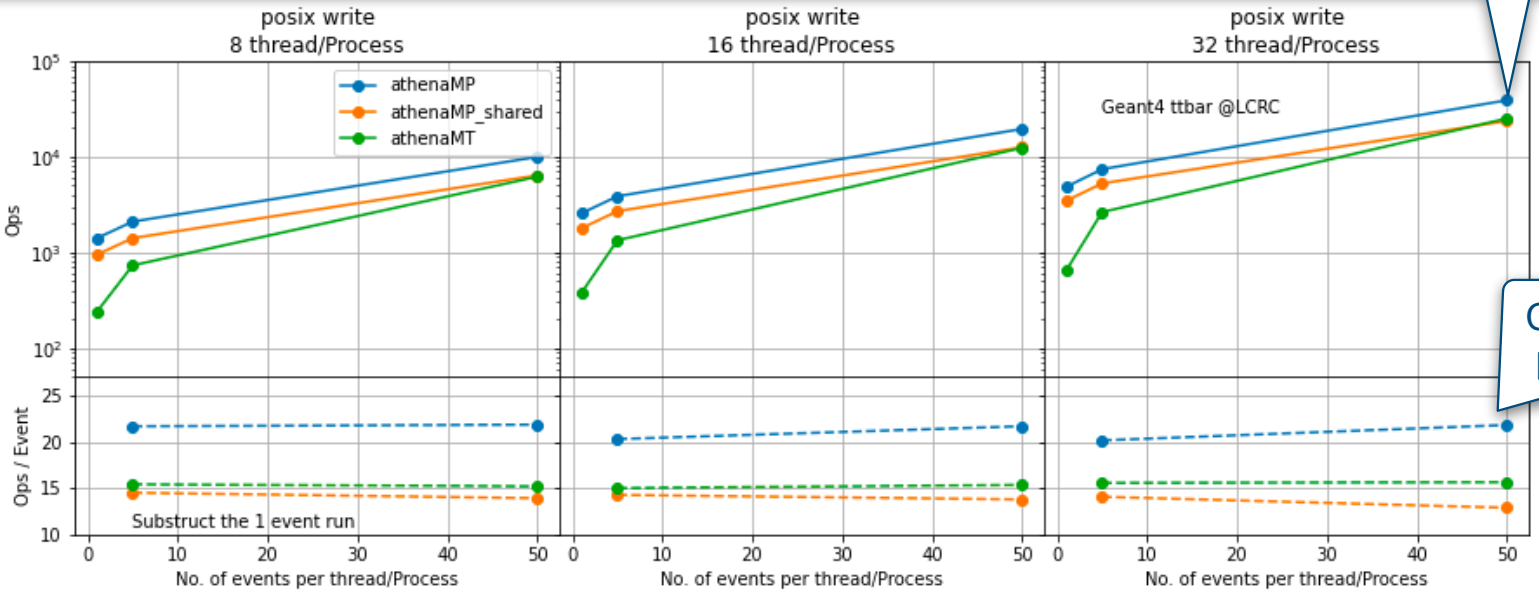
- In AthenaMP each worker does the writes while SharedWirter took over these



POSIX Operations

- In AthenaMP each worker does the write while SharedWirter took them over

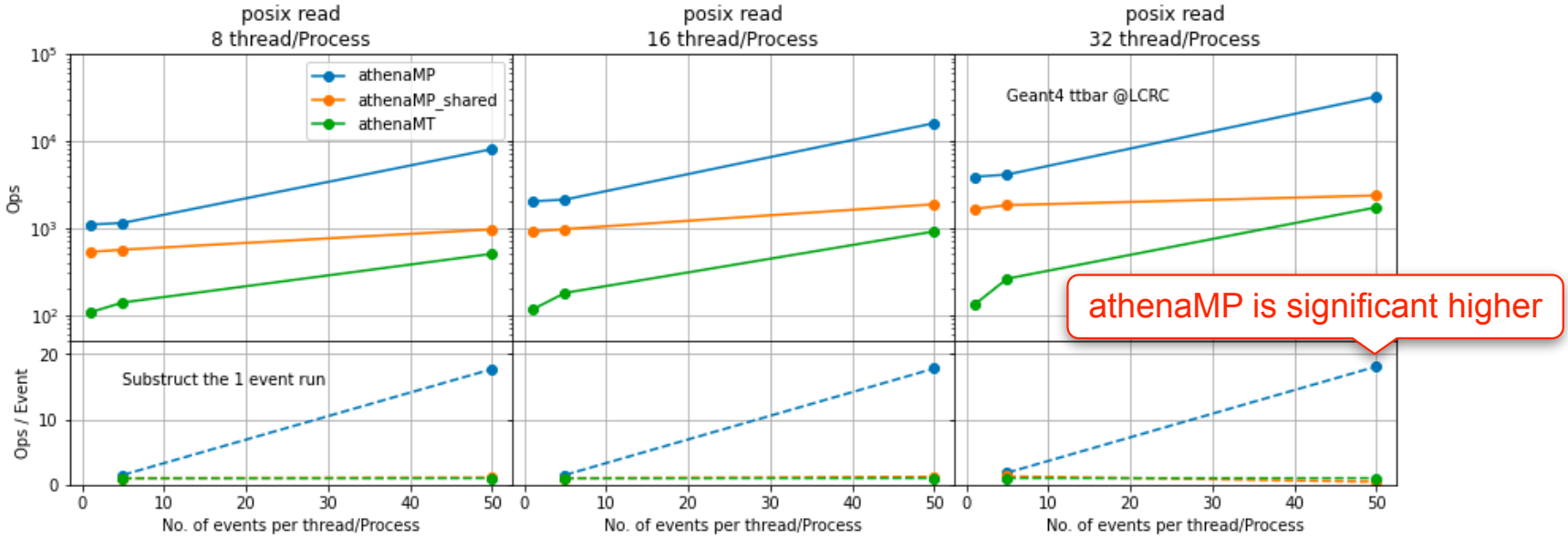
Increment mainly due to the additional merge process which combines the individual output files



Consistent per event

POSIX Operations

- The READ operations need further understanding



POSIX Operations

- The READ operations need further understanding
  - Difference in AthenaMP is dominated by the post-processing merge process
  - Because merge process has to read 'large' worker output

Mode	nproc	nevt	proc	posix read	posix write
athenaMP	16	1 / 50	main	99 / 99	54 / 54
athenaMP	16	1 / 50	merge	**891 / 13888**	295 / 5116
athenaMP	16	1 / 50	worker	61 / ~120	135 / ~900
athenaMP_shared	16	1 / 50	main	99 / 99	32 / 32
athenaMP_shared	16	1 / 50	shared_writer	84 / 84	1721 / 12521
athenaMP_shared	16	1 / 50	worker	43 / ~100	0 / 0
athenaMT	16	1 / 50	main	115 / 911	377 / 12405

Summary

- First look on the I/O activities of Athena
  - AthenaMP, AthenaMP+SharedWriter, AthenaMT
  - POSIX operations
- Understanding the patterns

- Using Darshan as the I/O monitoring tool for Atlas HPC workflow
  - Add Darshan to /CVMFS and ATLAS pilot script
  - Customized runtime exclusion and inclusion list for each production step