Performance Observability

Summer School
Sept. 2025

ddn

Jean-Thomas Acquaviva
jtacquaviva@ddn.com

Observability viewpoint

View point: Pros and Cons

COMPUTE NODES

Detailed I/O patterns

Detailed temporal behavior

Link to the source code

Increased I/O footprint accelerating computations

Overhead of observation

From single node to multiple nodes nodes

Ignorant of the layered IO stacks

SYSTEM WIDE LEVEL

Ease of observation (always-on)

Capture Interference

Coarse grain

Root access

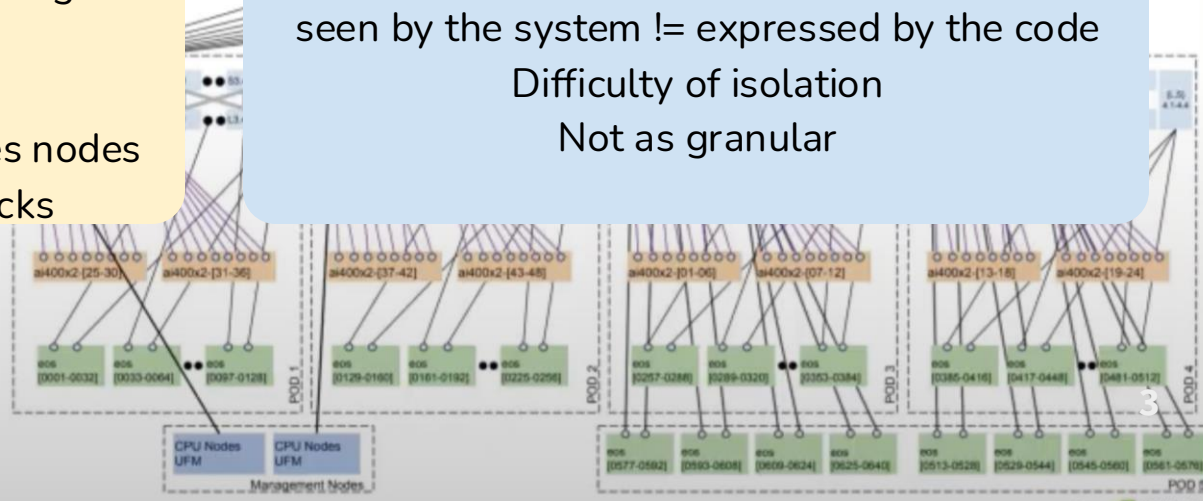
seen by the system != expressed by the code

Difficulty of isolation

Not as granular

IB

- Appliance IB connections are interleaved across 4 PODs
- Target a minimum performance of 2 TB/s (read) to support DL training at scale

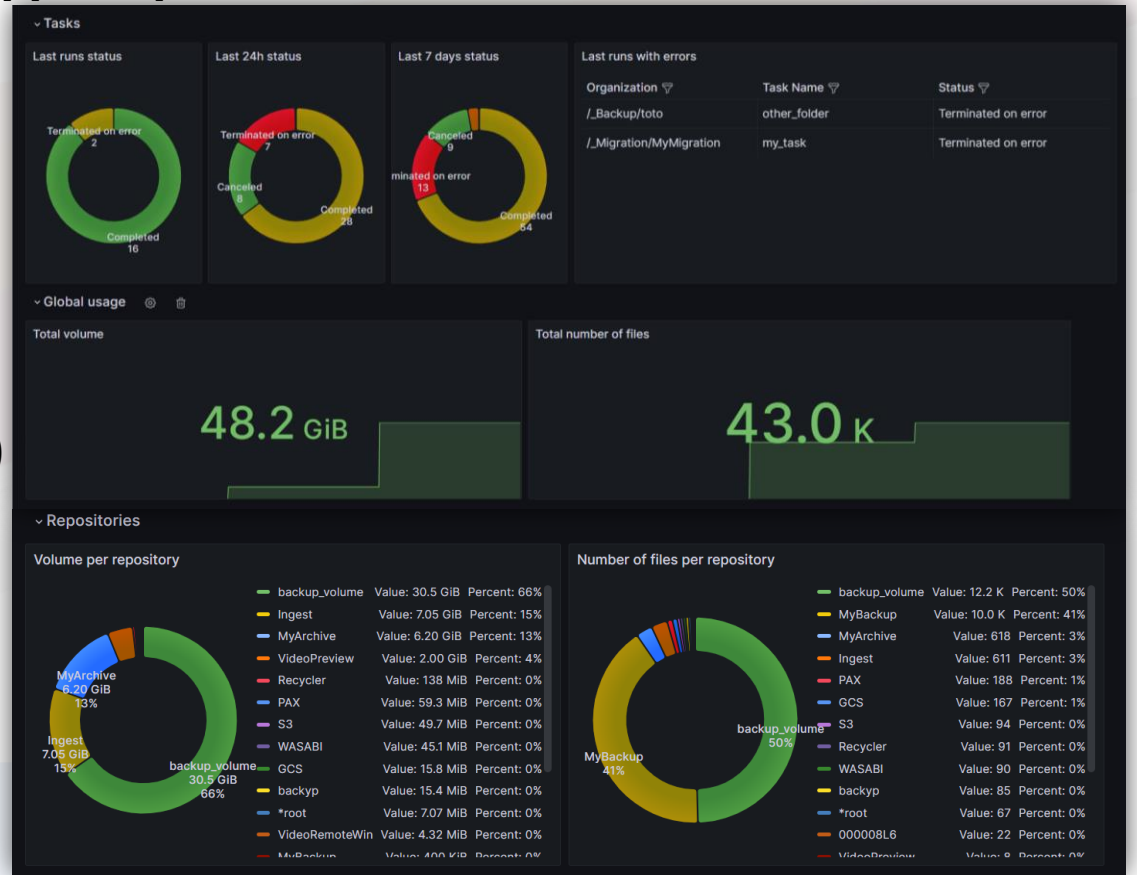


Observability from the File System

Lustre Jobstat

- Lustre (DDN's EXAScaler) supports probes
- OpenMetrics format
- Prometheus Integration
- Grafana display

- Scripting rules and Alert
- Glljobstats (DDN Eu. github)



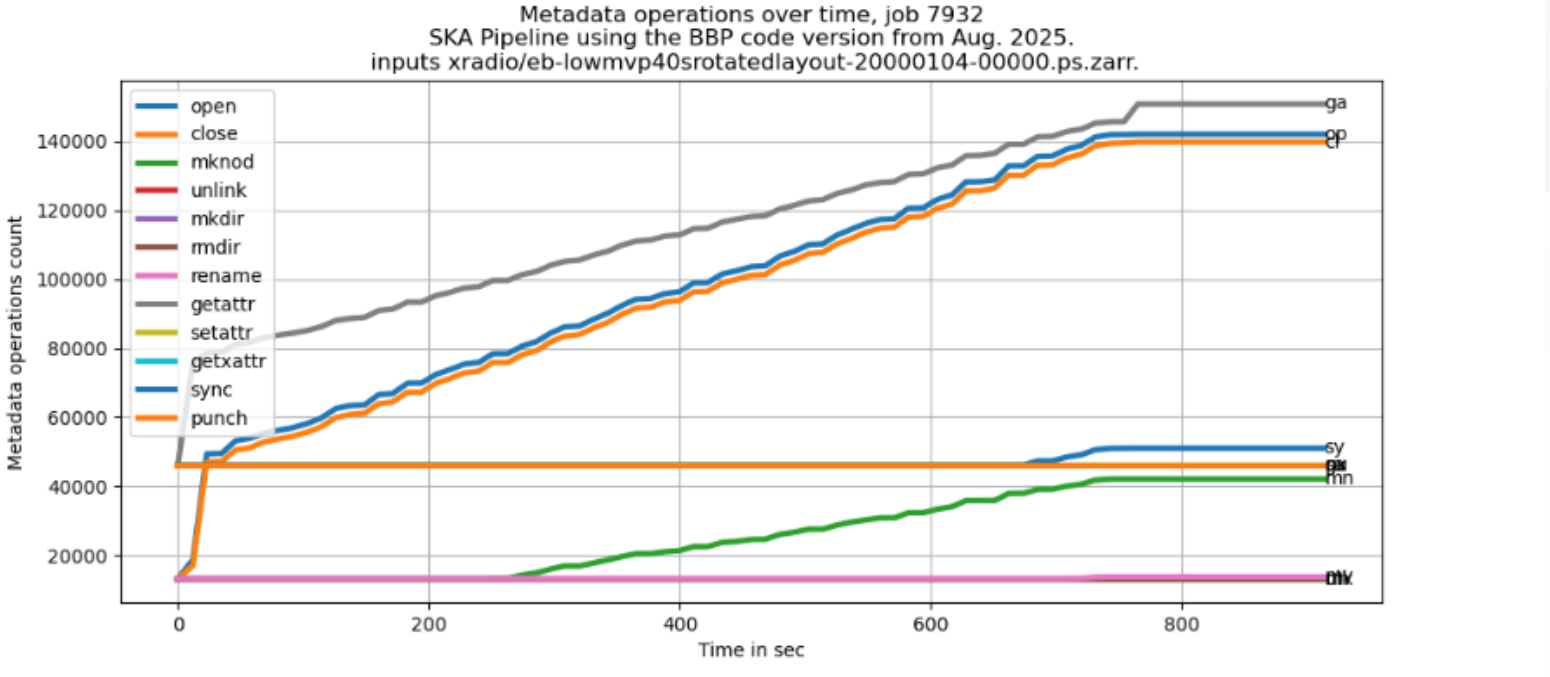
Code Characterizarion

System wide: all the IO traffic from all the nodes hitting EXAScaler is analyzed and reported

glljobstat.py [-h] [-cfg CONFIGFILE] [-c COUNT] [-i INTERVAL] [-n REPEATS] [--param PARAM] [--groupby GROUPBY] [--sortby SORTBY] [-o] [-m] [-s SERVERS]
[--fullname] [--no-fullname] [-f FILTER] [-fm] [-l JOBID_LENGTH] [-t] [-tr] [-minr MINRATE] [-trf TOTALRATEFILE] [-p] [-ht] [-nps NUM_PROC_SSH]

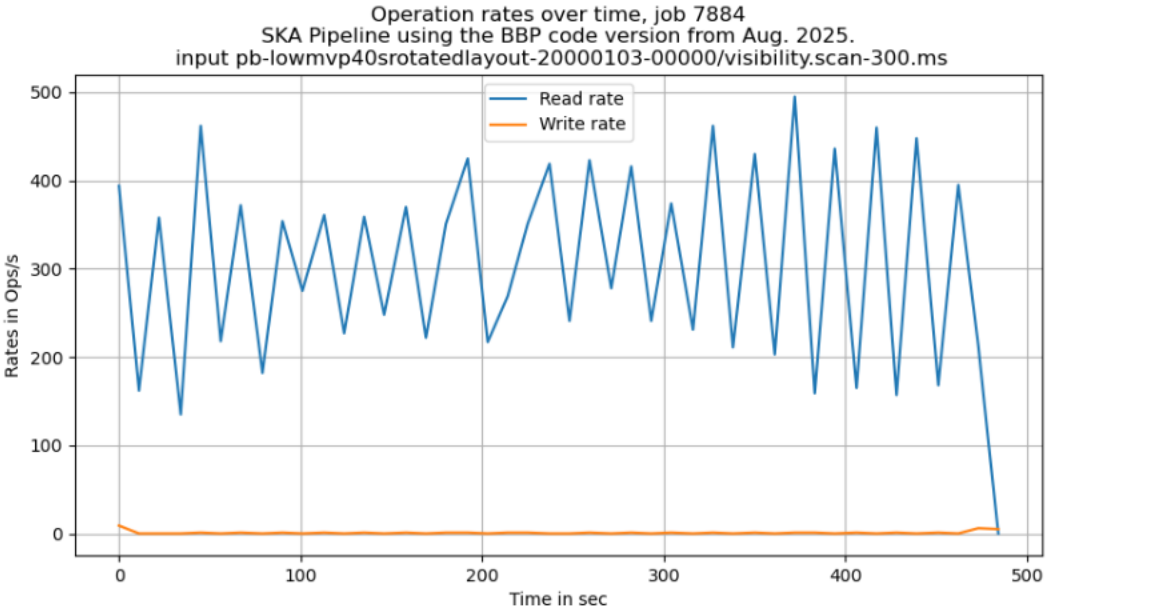
- [-npp NUM_PROC_DATA] [-ncs NUM_CHUNK_SSH] [-ncp NUM_CHUNK_DATA] [-hi] [-v] [-d | -r]
- -c COUNT, --count COUNT
- the number of top jobs to be listed (default 5).
- --param PARAM the param path to be checked (default *.*.job_stats).
- --groupby GROUPBY sort by **user / group / host / host_short / job / proc**

SKA Example: metadata operations rate



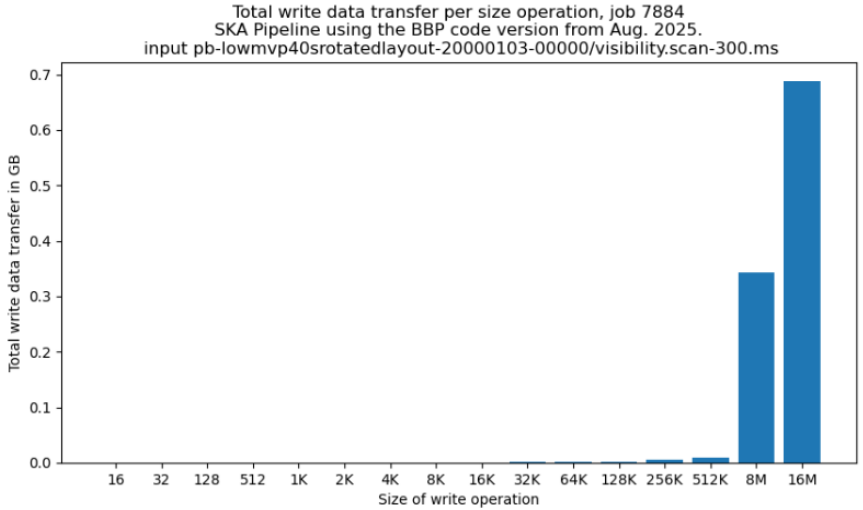
1. Assess the volume of metadata operation
2. Check temporal behavior
3. Check domination of specific metadata operation

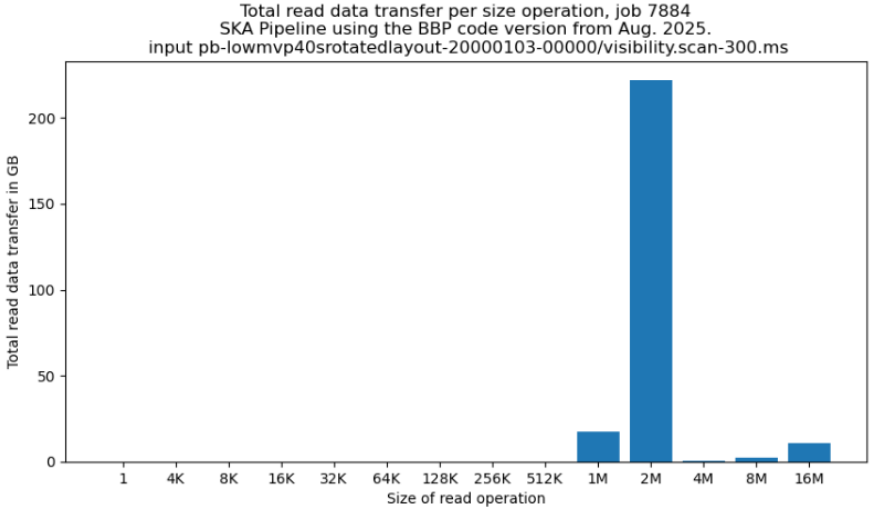
Data operations rate



1. Volume of data operations
2. Check temporal behavior (bursty vs steady)
3. Read / Write driven

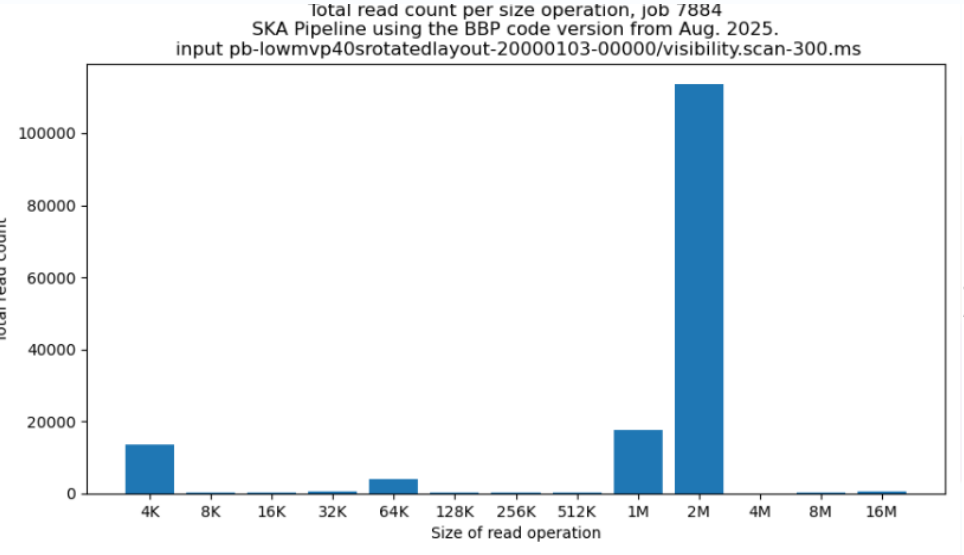
Data traffic: driving payload

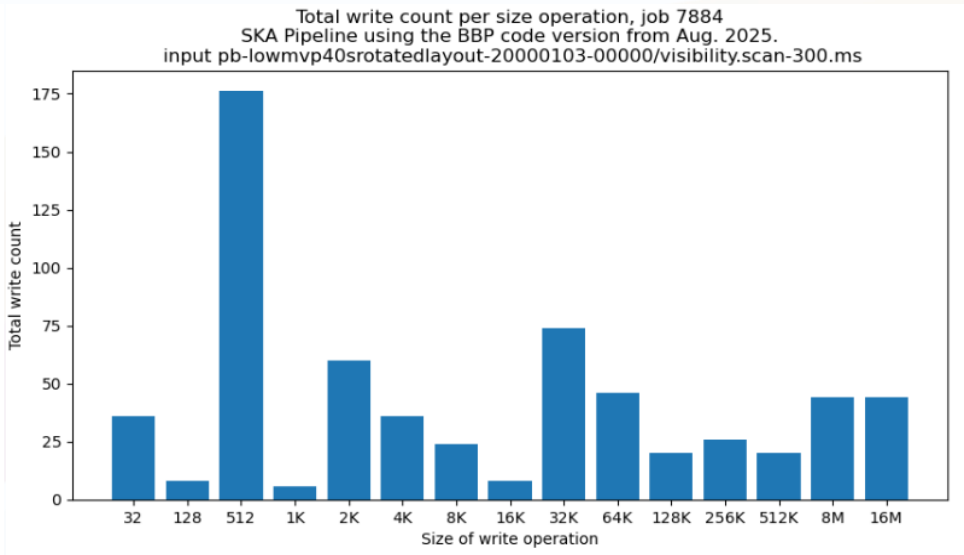




1. Payload size: > 1MB is preferable for network and storage efficiency
2. Large payload: good for hard drive

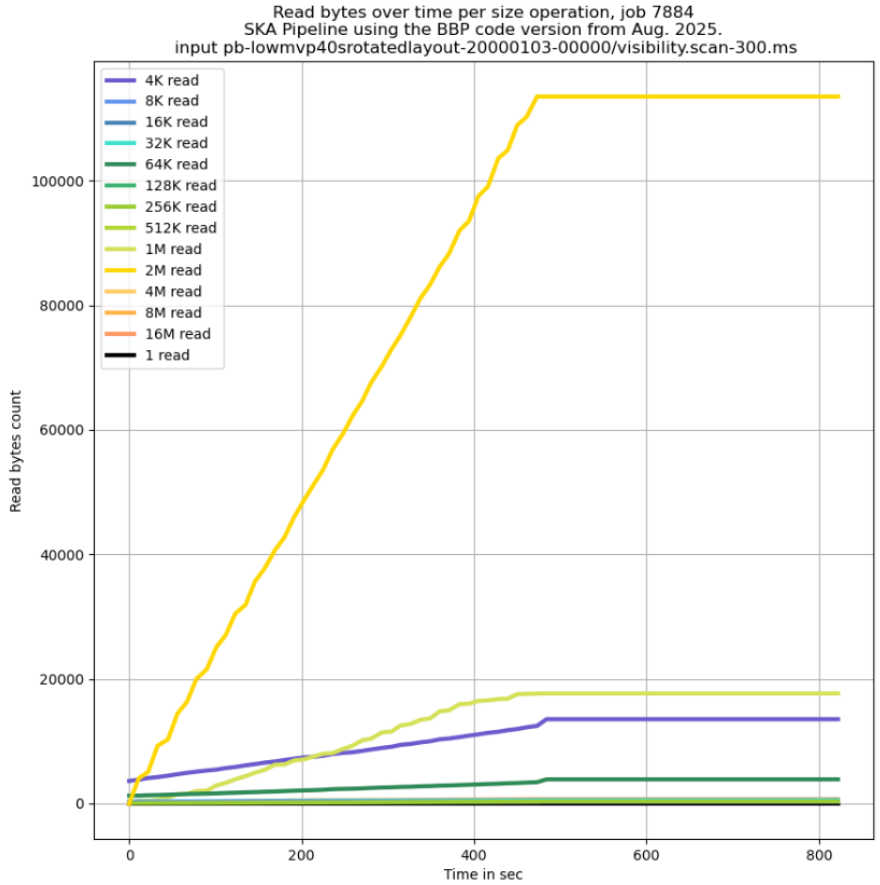
Data traffic: payload size





1. Small payload: use NVMe (latency driven)
2. Small payload: risk of serialization / locking
3. Performance impact of locking can drive application performance

Data payload Phase analysis



System wide Analysis

1. Ease of deployment
    - Monitoing and Profile
    - Limited overhead, light infrastructure
2. Job characterization, user characterization node characteization
    - Do the login/visualization nodes generate a lot of I/O traffic?
    - Is the user group "department X" consumming a lot of IO operations
3. Capture keys performance factors
    - Driving factor: payload, metadata operation
    - Temporal behavior
4. Overlooked some factors
    - Ranom vs sequential accesses
    - Shared vs file per object accessess

Observability from the Application

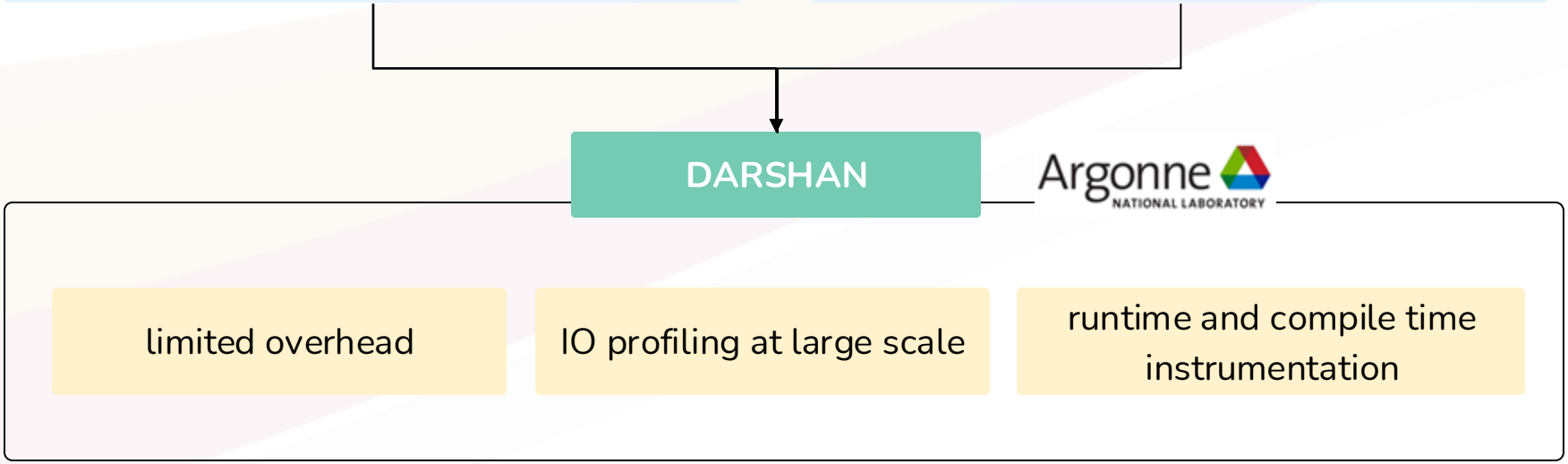
Detecting I/O bottlenecks with Darshan

Oct. 2024
JT. Acquaviva
jtacquaviva@ddn.com
Laura Bellentani
l.bellentani@cineca.it

Profiler Wish Lists

light
to support performance at large scale

modular
to easily integrated with new IO paradigms



https://github.com/darshan-hpc/darshan

Darshan

What can darshan do

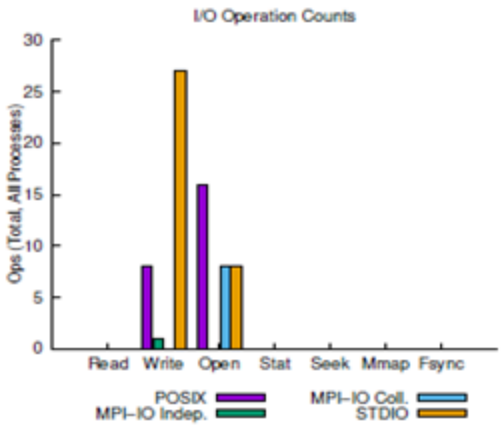
- Summaries of I/O (number of file, access size, total bytes written)
- DXT traces (call per timestamp and per rank)

Supported libraries

- POSIX
- STDIO
- MPI-IO

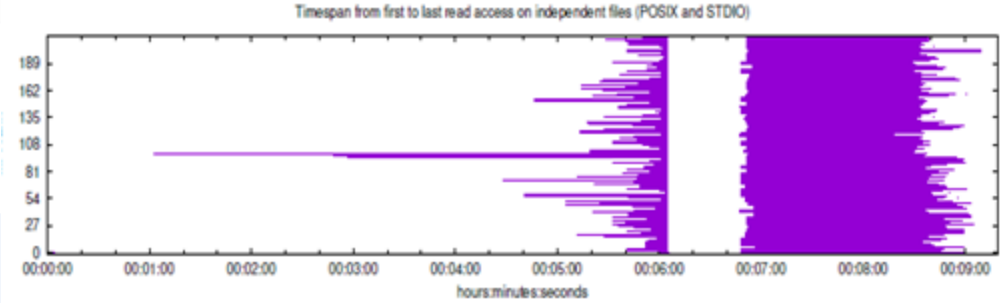
Recent additions

- HDF5
- PARALLEL NETCDF
- NON-MPI JOBS



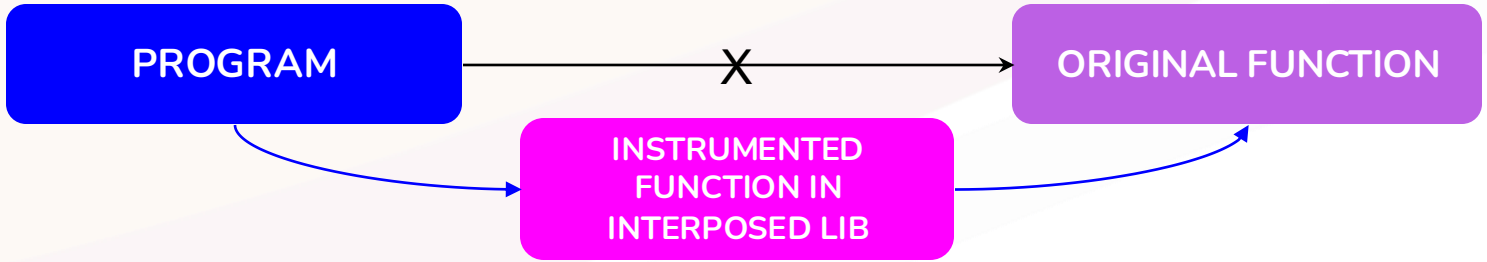
Variance in Shared Files (POSIX and STDIO)

File Suffix	Processes	Fastest			Slowest			σ	
		Rank	Time	Bytes	Rank	Time	Bytes	Time	Bytes
...misterio.out	16	14	0.299647	239MiB	2	2.165057	239MiB	0.597	0
...<STDOUT>	16	6	0.000010	149B	0	0.000034	255B	0	25.4



Instrumentation

- Library interposition with LD_PRELOAD for dynamic executable
  - Interceps I/O calls at runtime



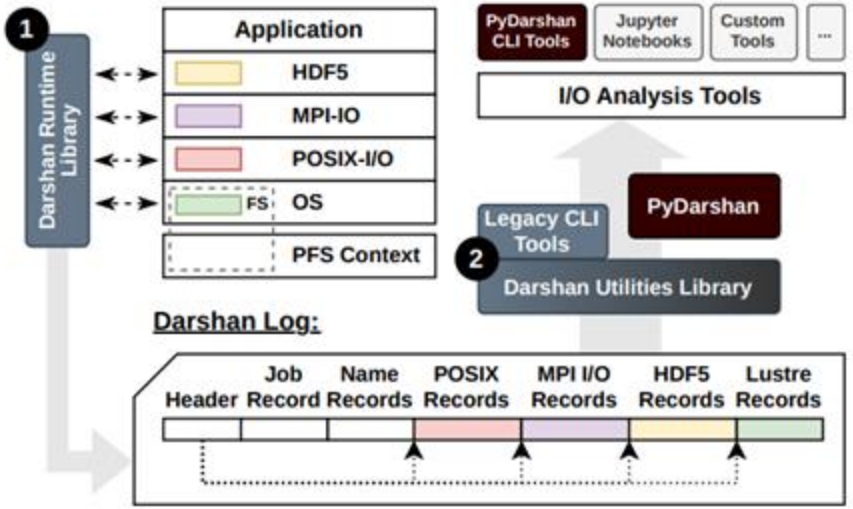
- insert instrumentation at compile time

```
subroutine mywrite()
start probe()
…
stop probe()
end subroutine
```

Instrumentation

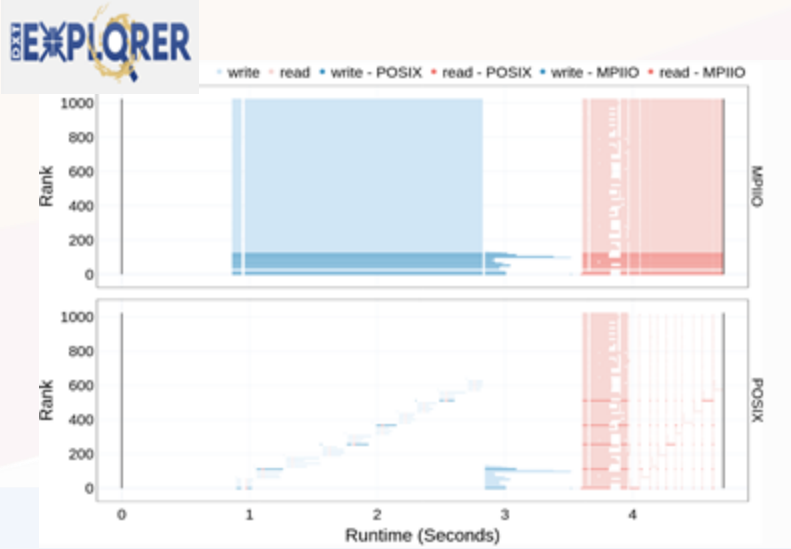
darshan-runtime

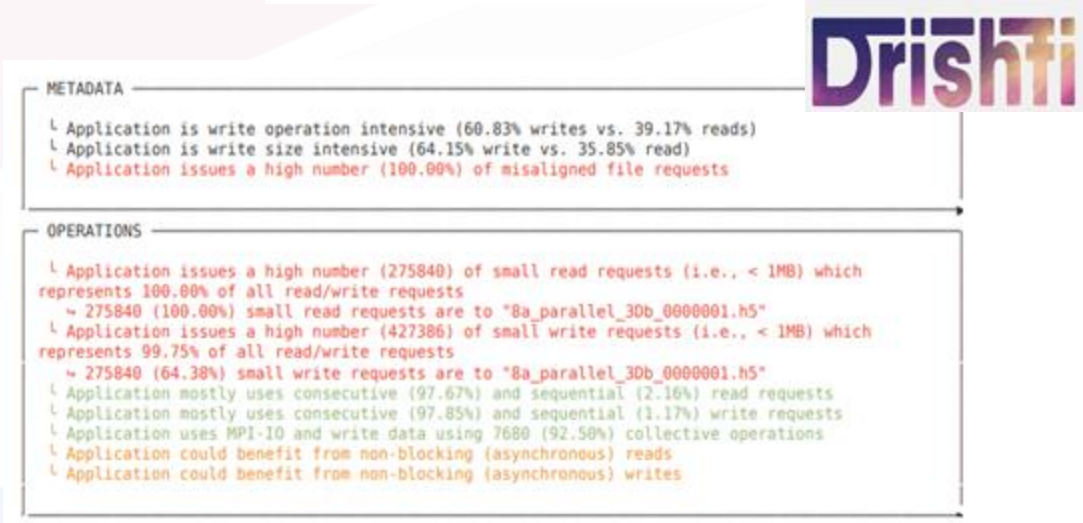
- Darshan starts/stops collection with MPI_Init and MPI_Finalize
- Information is collected per process and per rank
- Data is collected at runtime into buffers, aggregated and written in log files at app shutdown
- Postprocessing is done in a second stage



Postprocessing

- **darshan-util** : pdf with plots and tables
- **pydarshan** : heatmaps and HDF5
- **dxt_explorer** : interactive plots
- **Drishti** : feedbacks and hints





Darshan installation

REQUIREMENTS

- Configured with the same MPI library used for your application
- Installed with GCC compiler
- If using HDF5 and NetCDF, darshan configuration requires same library
- `darshan-util` requires `epstopdf, gnuplot, perl, python`
  - Latex needs to be in the path to generate reports with darshan-util

- SCC: on Future Technology Platform (FTP)
  - Darshan-util s available as a module (but not darshan-runtime)

```
[hpctraining32@login ~]$ module available

---------------------------------------------------------------------- /sw/modules/linux-rocky8-x86_64/Core ----------------------------------
   amdblis/4.1    amdlibflame/4.1    aocl-utils/4.1    apptainer/1.1.9    darshan-util/3.4.4    miniforge3/4.8.3-4-Linux-x86_64    openmpi/4.1.6
```

Darshan installation

AUTOTOOLS

```
tar -xvzf darshan-<version-number>.tar.gz
./prepare
cd darshan-<version-number>/darshan-runtime
mkdir -p build && cd build
../configure --with-jobid-env=SLURM_JOB_ID \
             --with-mem-align=8 \
             --enable-mmap-logs \
             --prefix=$DARSHAN_INSTALL_PATH \
             --with-log-path=<darshan log path> CC=mpicc
make install
```

SPACK

```
spack install darshan-runtime+mpi+hdf5+parallelnetcdf
spack install darshan-util
```

```
./darshan_setup.sh
```

darshan-runtime

```
LD_PRELOAD=$DARSHAN_INSTALL_PATH/lib/libdarshan.so <mpi launcher> <args> <executable>
```

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=32
#SBATCH --cpus-per-task=1
#SBATCH --time=00:20:00
#SBATCH --exclusive
#SBATCH --partition=boost_usr_prod
#SBATCH -q boost_qos_dbg

module purge
module load darshan-runtime/3.4.5--intel-oneapi-mpi--2021.10.0--gcc--12.2.0

export RUNID=$(date +"%Y%m%d%H%M")
export DARSHAN_INSTALL_PATH=$DARSHAN_RUNTIME_HOME
export DARSHAN_DUMP_CONFIG=1
export DARSHAN_LOGPATH=$PWD
export DXT_ENABLE_IO_TRACE=1
export DARSHAN_MODMEM=4000

export DARSHAN_LOGFILE=$DARSHAN_LOGPATH/darshan.log
LD_PRELOAD=$DARSHAN_INSTALL_PATH/lib/libdarshan.so mpiexec -np 9 ../bin/bt.B.16.mpi_io_full
```

darshan-runtime

Export environment variables to customize your measurement

`DARSHAN_LOGFILE` → set the name and directory of the logfile

`DARSHAN_MODMEM` → used to increase the size of darshan buffer

`DARSHAN_DXT_ENABLE` → enables tracing

`DARSHAN_DISABLE_SHARED_REDUCTION` → quench reduction of stats over MPI ranks for shared

`DARSHAN_DUMP_CONFIG` → prints configuration options to stderr/stdout

Configure the measurement

```
##########################
##### DARSHAN CONFIG #####
##########################
# MODMEM = 4000 MiB
# NAMEMEM = 1024 KiB
# MEM_ALIGNMENT = 8 bytes
# JOBID = NONE
# LOGHINTS = romio_no_indep_rw=true;cb_nodes=4
# LOGPATH = /leonardo/home/userinternal/lbellen1/work/projects/profiling-support/darshan/space-event/booster/NPB3.3.1/NPB3.3-MPI/darshan
# LOGPATH_BYENV = /leonardo/home/userinternal/sgiulian
# EXCLUDE_DIRS = /etc/,/dev/,/usr/,/bin/,/boot/,/lib/,/opt/,/sbin/,/sys/,/proc/,/var/
# APP_EXCLUDE = NONE
# APP_INCLUDE = NONE
# RANK_EXCLUDE = NONE
# RANK_INCLUDE = NONE
# POSIX MODULE CONFIG:
#       - ENABLED
# MPI-IO MODULE CONFIG:
#       - ENABLED
# H5F MODULE CONFIG:
#       - ENABLED
# H5D MODULE CONFIG:
#       - ENABLED
# PNETCDF_FILE MODULE CONFIG:
#       - ENABLED
# PNETCDF_VAR MODULE CONFIG:
#       - ENABLED
# BG/Q MODULE CONFIG:
#       - DISABLED
# LUSTRE MODULE CONFIG:
#       - ENABLED
# STDIO MODULE CONFIG:
#       - ENABLED
# DXT_POSIX MODULE CONFIG:
#       - ENABLED
# DXT_MPIIO MODULE CONFIG:
#       - ENABLED
# MDHIM MODULE CONFIG:
#       - DISABLED
# APXC MODULE CONFIG:
#       - DISABLED
# APMPI MODULE CONFIG:
#       - DISABLED
# HEATMAP MODULE CONFIG:
#       - ENABLED
##########################
### END DARSHAN CONFIG ###
```

DARSHAN_DUMP_CONFIG=1

measure only selected applications workflows

instrument only specific ranks analyze load imbalances

exclude some filesystem directories MPI implementations might generate temporary files, expensive for tracing

enable/disable HDF5 and ParallelNetCDF

enable/disable DXT and heatmaps to reduce the overhead when needed

Configure the measurement

In addition to exporting environment variables, a configuration file with more options can be used for DARSHAN_CONFIG_PATH

```
# 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     4096      POSIX,MPI-IO

# 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    ^/home        *
NAME_INCLUDE    .out$         *

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

# avoid generating logs for git and ls binaries
APP_EXCLUDE     git,ls

# exclude instrumentation for all ranks first
RANK_EXCLUDE    0:
# then selectively re-include ranks 0-3 and 12:15
RANK_INCLUDE    0:3
```

Pay attention ! It could prevent instrumentation of the files your're looking for!

Summaries

Generates pdf with graphs and tables of the summarized metrics to streamline the identification of bottlenecks

```
$ darshan-job-summary <darshan-logfile>
```

Not working on FTP :-(

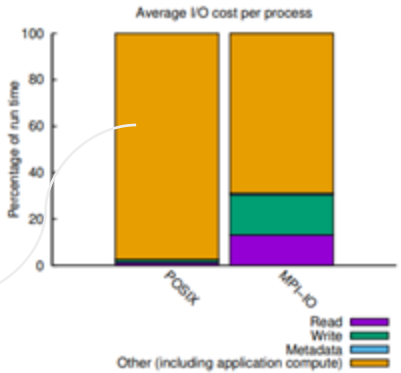
jobid: 1523575	uid: 33678	nprocs: 9	runtime: 6.4439 seconds

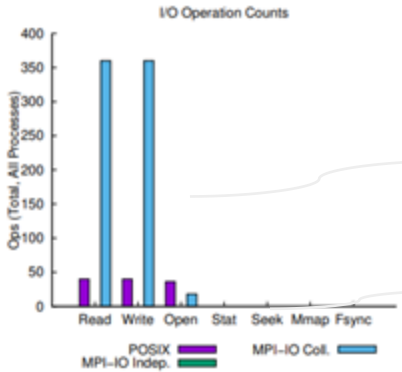
I/O performance *estimate* (at the MPI-IO layer): transferred 800.0 MiB at 394.00 MiB/s

compare the time spent in different POSIX/STDIO/MPI-IO operations

identify the I/O footprint of your application

understand how much relevant is IO in your workload (and which module)





For MPI-IO distinguish between collective and independent (more expensive)

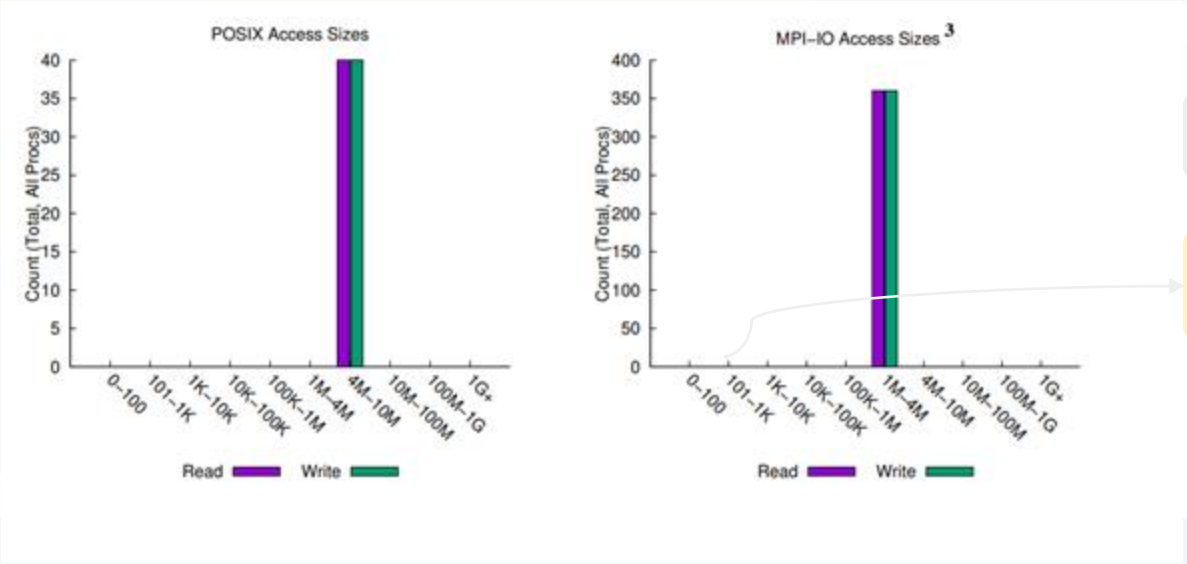
consider replacing independent MPI IO with collective

identify potentially expensive IO operations (seek)

Summaries

Generates pdf with graphs and tables of the summarized metrics to streamline the identification of bottlenecks

```
$ darshan-job-summary <darshan-logfile>
```



Identify size of accesses per module and per operation

Useful to identify potential bottlenecks due to small access sizes!

Summaries

Generates pdf with graphs and tables of the summarized metrics to streamline the identification of bottlenecks

```
$ darshan-job-summary <darshan-logfile>
```

see the average access size

Most Common Access Sizes (POSIX or MPI-IO)

	access size	count
POSIX	10485760	80
MPI-IO ‡	1165080	480
	1164240	160
	1166800	80

NOTE: MPI-IO accesses are given in terms of aggregate datatype size.

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

type	number of files	avg. size	max size
total opened	1	400MiB	400MiB
read-only files	0	0	0
write-only files	0	0	0
read/write files	1	400MiB	400MiB
created files	1	400MiB	400MiB

Tables show the time spent in metadata versus reads or write for independent or shared files

Identify which filesystem is used and how much data per filesystem

Average I/O per process (POSIX and STDIO)

	Cumulative time spent in I/O functions (seconds)	Amount of I/O (MiB)
Independent reads	0	0
Independent writes	0	0
Independent metadata	0	N/A
Shared reads	0.0752848888888889	44.4444444444444
Shared writes	0.087403	44.4444444444444
Shared metadata	0.001337	N/A

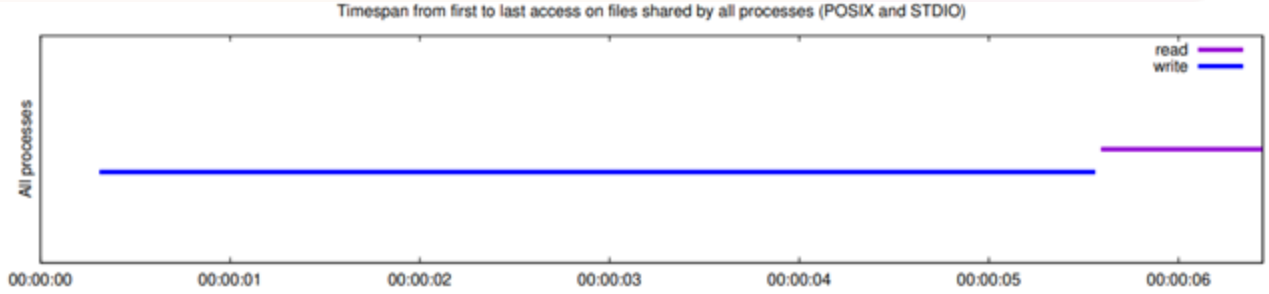
Data Transfer Per Filesystem (POSIX and STDIO)

File System	Write		Read	
	MiB	Ratio	MiB	Ratio
/leonardo_work	400.00000	1.00000	400.00000	1.00000

Summaries

Generates pdf with graphs and tables of the summarized metrics to streamline the identification of bottlenecks

```
$ darshan-job-summary <darshan-logfile>
```



Variance in Shared Files (POSIX and STDIO)

File Suffix	Processes	Fastest			Slowest			σ	
		Rank	Time	Bytes	Rank	Time	Bytes	Time	Bytes
...tio.full.out	9	2	0.000793	0	0	1.465457	800MiB	0.46	2.64e+08

Timespan of accesses on shared files (same file are reduced over MPI ranks)

in reduction mode can not see the access to shared files

Consider using export DARSHAN_DISABLE_SHARED_REDUCTION

darshan-util

Provides CLIs to postprocess the darshan logfiles and display summaries/traces in human readable format

```
$ darshan-parser <darshan-logile>
```

```
# darshan log version: 3.41
# compression method: ZLIB
# exe: ../bin/bt.A.9.mpi_io_full
# uid: 33678
# jobid: 1523575
# start_time: 1721046257
# start_time_asci: Mon Jul 15 14:24:17 2024
# end_time: 1721046264
# end_time_asci: Mon Jul 15 14:24:24 2024
# nprocs: 9
# run time: 6.4439
# metadata: lib_ver = 3.4.5
# metadata: h = romio_no_indep_rw=true;cb_nodes=4

# log file regions
# -------------------------------------------------------
# header: 1328 bytes (uncompressed)
# job data: 296 bytes (compressed)
# record table: 172 bytes (compressed)
# POSIX module: 179 bytes (compressed), ver=4
# MPI-IO module: 186 bytes (compressed), ver=3
# LUSTRE module: 37 bytes (compressed), ver=1
# DXT_POSIX module: 1011 bytes (compressed), ver=1
# DXT_MPIIO module: 9469 bytes (compressed), ver=2
# HEATMAP module: 2281 bytes (compressed), ver=1
```

summary

how many bytes measured per module

```
# mounted file systems (mount point and fs type)
# -------------------------------------------------------
# mount entry:  /sys/firmware/efi/efivars       efivarfs
# mount entry:  /sys/kernel/debug/tracing       tracefs
# mount entry:  /proc/sys/fs/binfmt_misc        autofs
# mount entry:  /sys/kernel/tracing     tracefs
# mount entry:  /sys/kernel/config      configfs
# mount entry:  /leonardo_scratch       lustre
# mount entry:  /sys/fs/pstore  pstore
# mount entry:  /leonardo_work  lustre
# mount entry:  /sys/fs/bpf     bpf
# mount entry:  /dev/mqueue     mqueue
# mount entry:  /opt/intel      lustre
# mount entry:  /leonardo       lustre
# mount entry:  /opt/cuda       lustre
# mount entry:  /etc/nhc        lustre
# mount entry:  /homeard        lustre
# mount entry:  /dev    devtmpfs
# mount entry:  /       ext4
```

filesystem availables

darshan-util

Provides CLIs to postprocess the darshan logfiles and display summaries/traces in human readable format

```
$ darshan-parser <darshan-logile>
```

summary of counters for the different modules enabled, files (name and hashes), mount point, fs type…

-1 if file is shared and stats are aggregated ; use DARSHAN_DISABLE_SHARED_REDUCTION

```
# description of columns:
#   <module>: module responsible for this I/O record.
#   <rank>: MPI rank.  -1 indicates that the file is shared
#      across all processes and statistics are aggregated.
#   <record id>: hash of the record's file path
#   <counter name> and <counter value>: statistical counters.
#      A value of -1 indicates that Darshan could not monitor
#      that counter, and its value should be ignored.
#   <file name>: full file path for the record.
#   <mount pt>: mount point that the file resides on.
#   <fs type>: type of file system that the file resides on.

#<module>       <rank>  <record id>     <counter>       <value> <file name>     <mount pt>      <fs type>
MPI-IO  -1      9704632873208484545     MPIIO_INDEP_OPENS       0       /leonardo_work/cin_staff/lbellen1/projects/profiling-support/darsh
an/space-event/booster/NPB3.3.1/NPB3.3-MPI/darshan/btio.full.out        /leonardo_work  lustre
MPI-IO  -1      9704632873208484545     MPIIO_COLL_OPENS        18      /leonardo_work/cin_staff/lbellen1/projects/profiling-support/darsh
an/space-event/booster/NPB3.3.1/NPB3.3-MPI/darshan/btio.full.out        /leonardo_work  lustre
MPI-IO  -1      9704632873208484545     MPIIO_INDEP_READS       0       /leonardo_work/cin_staff/lbellen1/projects/profiling-support/darsh
an/space-event/booster/NPB3.3.1/NPB3.3-MPI/darshan/btio.full.out        /leonardo_work  lustre
MPI-IO  -1      9704632873208484545     MPIIO_INDEP_WRITES      0       /leonardo_work/cin_staff/lbellen1/projects/profiling-support/darsh
an/space-event/booster/NPB3.3.1/NPB3.3-MPI/darshan/btio.full.out        /leonardo_work  lustre
MPI-IO  -1      9704632873208484545     MPIIO_COLL_READS        360     /leonardo_work/cin_staff/lbellen1/projects/profiling-support/darsh
an/space-event/booster/NPB3.3.1/NPB3.3-MPI/darshan/btio.full.out        /leonardo_work  lustre
MPI-IO  -1      9704632873208484545     MPIIO_COLL_WRITES       360     /leonardo_work/cin_staff/lbellen1/projects/profiling-support/darsh
an/space-event/booster/NPB3.3.1/NPB3.3-MPI/darshan/btio.full.out        /leonardo_work  lustre
MPI-IO  -1      9704632873208484545     MPIIO_SPLIT_READS       0       /leonardo_work/cin_staff/lbellen1/projects/profiling-support/darsh
an/space-event/booster/NPB3.3.1/NPB3.3-MPI/darshan/btio.full.out        /leonardo_work  lustre
```

darshan-util

Provides CLIs to postprocess the darshan logfiles and display summaries/traces in human readable format

```
$ darshan-parser --total <darshan-logile>

total_MPIIO_INDEP_OPENS: 0
total_MPIIO_COLL_OPENS: 18
total_MPIIO_INDEP_READS: 0
total_MPIIO_INDEP_WRITES: 0
total_MPIIO_COLL_READS: 360
total_MPIIO_COLL_WRITES: 360
total_MPIIO_SPLIT_READS: 0
total_MPIIO_SPLIT_WRITES: 0
total_MPIIO_NB_READS: 0
total_MPIIO_NB_WRITES: 0
total_MPIIO_SYNCS: 0
total_MPIIO_HINTS: 0
total_MPIIO_VIEWS: 18
total_MPIIO_MODE: 2
total_MPIIO_BYTES_READ: 419430400
total_MPIIO_BYTES_WRITTEN: 419430400
total_MPIIO_RW_SWITCHES: 9
total_MPIIO_MAX_READ_TIME_SIZE: 1164240
total_MPIIO_MAX_WRITE_TIME_SIZE: 1165080
```

total number of events, summarized over ranks and files

get a general overview of IO characterization and potential bottlenecks

darshan-util

Provides CLIs to postprocess the darshan logfiles and display summaries/traces in human readable format

```
$ darshan-parser --perf <darshan-logile>

# ***************************************************
# MPI-IO module data
# ***************************************************

# performance
# -----------
# total_bytes: 838860800
#
# I/O timing for unique files (seconds):
# ......................
# unique files: slowest_rank_io_time: 0.000000
# unique files: slowest_rank_meta_only_time: 0.000000
# unique files: slowest_rank_rw_only_time: 0.000000
# unique files: slowest_rank: 0
#
# I/O timing for shared files (seconds):
# ......................
# shared files: time_by_slowest: 2.030464
#
# Aggregate performance, including both shared and unique files:
# ......................
# agg_time_by_slowest: 2.030464 # seconds
# agg_perf_by_slowest: 393.998533 # MiB/s
```

performance (bandwidth) for unique and shared files

DXT traces

export DXT_ENABLE_IO_TRACE=1

darshan-dxt-parser <darshan-log> > darshan.dxt.out

identify the size of POSIX read / write operations

identify the offset of sequential read / write operation to see if these are consecutive

```
# ************************************************
# DXT_POSIX module data
# ************************************************

# DXT, file_id: 4587689819065279387, file_name: /leonardo_work/cin_staff/lbellen1/projects/profiling-support/darshan/space-event/booster/grco-7k/SAVE/ns.db1
# DXT, rank: 0, hostname: lrdn1360.leonardo.local
# DXT, write_count: 0, read_count: 153
# DXT, mnt_pt: /leonardo_work, fs_type: lustre
# DXT, Lustre stripe_size: 1048576, Lustre stripe_count: 1
# DXT, Lustre OST obdidx: 54
# Module    Rank  Wt/Rd  Segment          Offset       Length    Start(s)      End(s)  [OST]
 X_POSIX       0   read        0               0            8      2.0136      2.0136  [ 54]
 X_POSIX       0   read        1               0           16      2.0142      2.0142  [ 54]
 X_POSIX       0   read        2              16           32      2.0142      2.0142  [ 54]
 X_POSIX       0   read        3              48          512      2.0163      2.0163  [ 54]
 X_POSIX       0   read        4            9967           38      2.0170      2.0170  [ 54]
 X_POSIX       0   read        5            9522          146      2.0172      2.0172  [ 54]
 X_POSIX       0   read        6           10882          512      2.0181      2.0181  [ 54]
 X_POSIX       0   read        7            8771           53      2.0185      2.0185  [ 54]
 X_POSIX       0   read        8           15275          512      2.0185      2.0185  [ 54]
```

DXT traces

Well known issue with DXT and Lustre module:

```
$ darshan-dxt-parser nas-bt-B-16.darshan-noreduction
Error: failed to parse LUSTRE module record.
```

Use a configuration file to exclude the module from instrumentation

```
MAX_RECORDS     65536      POSIX,MPI-IO,STDIO #adjust and increase based on your modules
MODMEM  1024    #you can increase these numbers based on your application
MOD_DISABLE LUSTRE
```

! PROFILING I/O on Booster and DCGP
ompio I/O plugin (OpenMPI) generates *locktest* temporary files when accessing shared files. DXT traces might exceed available buffer. Consider using:

```
NAME_EXCLUDE        locktest  *
```

Intel-oneapi-mpi is based on **romio MPI I/O (MPICH)**, which does not generate these explicit tmp files.

Pydarshan

What Pydarshan provides

- html reports
- APIs to collect custom analysis

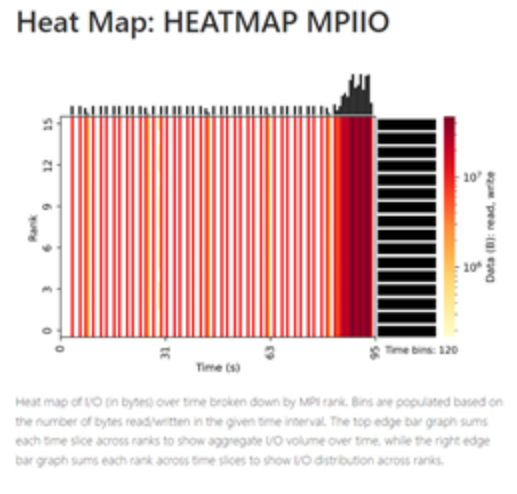
Why Pydarshan

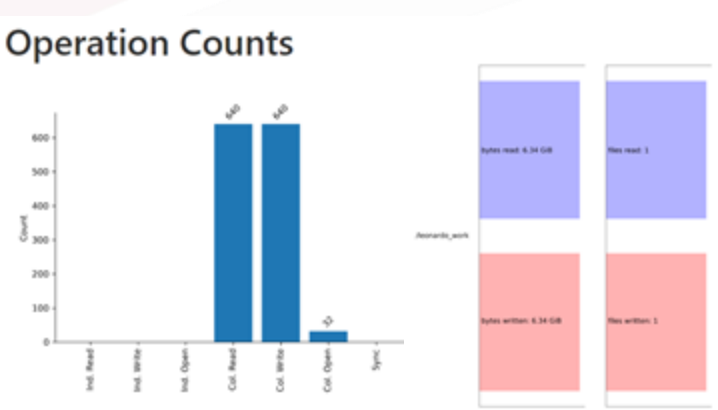
- Connect Darshan performance data to a rich Python ecosystem of data science, machine learning, and visualization libraries
- Promote interoperability of performance analysis tools by facilitating access to reusable analysis routines
- Enable more efficient access to Darshan data for the analysis of large collections of Darshan logs and longitudinal studies

Pydarshan

Complementary to darshan-util, displays also heatmaps and dxt traces

```
python summary.py <darshan-logfile> [--enable_dxt_heatmap] --output <outputname>
```





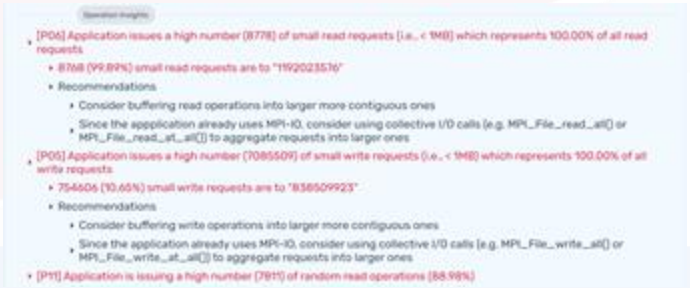
HPCIO Analysis Platform

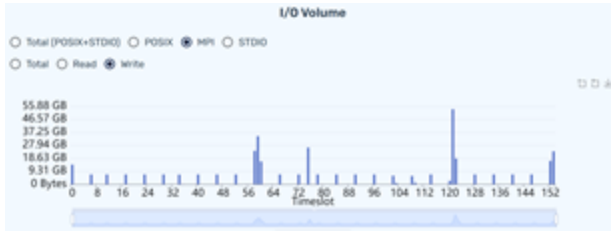
https://hpcioanalysis.zdv.uni-mainz.de/

- database of I/O performance of applications running on HPC clusters
- collaborative effort to improve knowledge in I/O aspects in HPC, e.g. metadata operations and common I/O access patterns
- foster the design of new storage solution for the HPC world and related applications

Upload darshan logs with DXT enabled → returns detailed analysis integrated with Drishti!

Example : NEK5000





Thank you!

Thanks to:

- **Laura Bellentani**
- Darshan developers
- Nafiseh Moti from Leibniz supercomputing