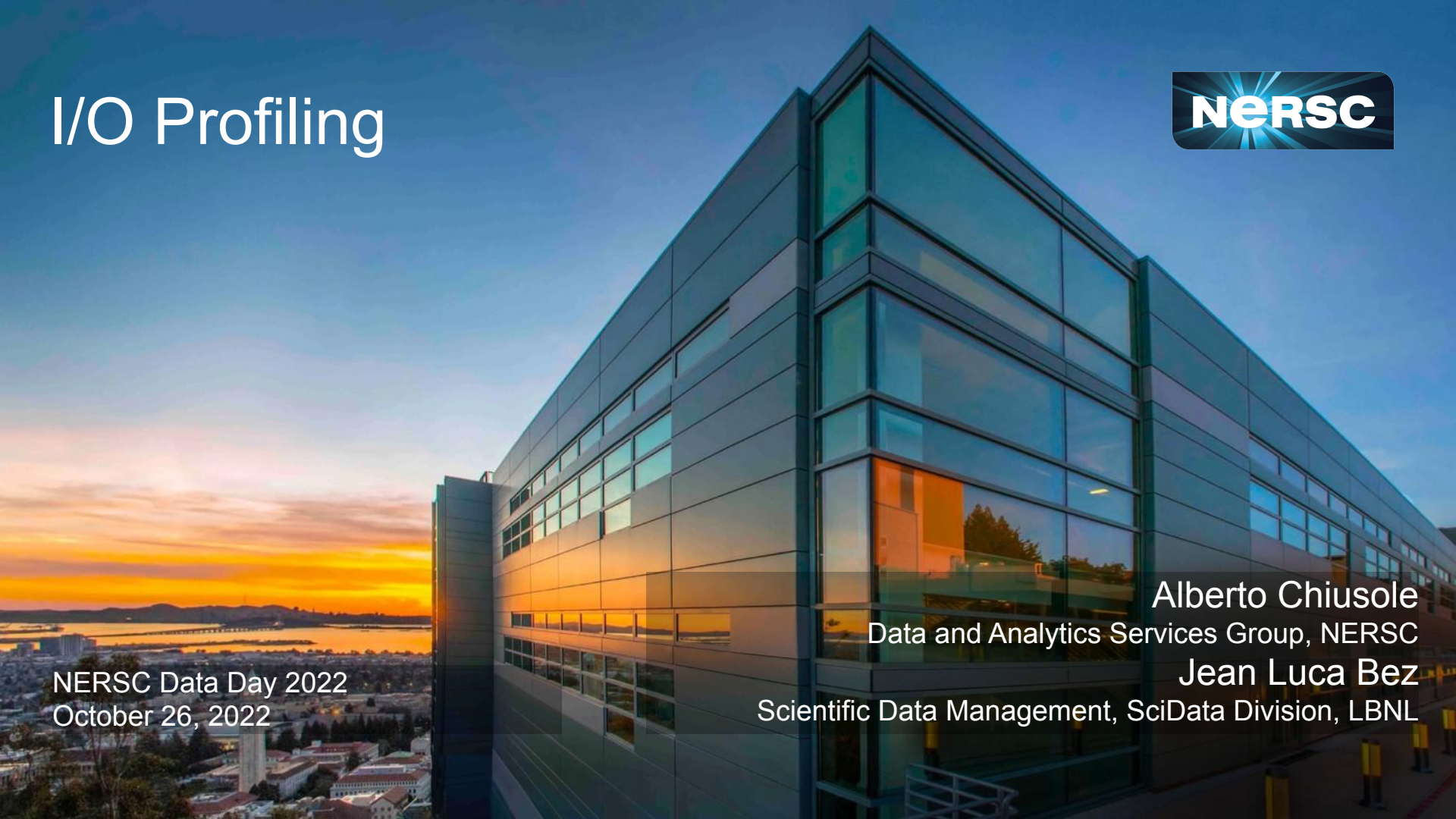
I/O Profiling



Alberto Chiusole
Data and Analytics Services Group, NERSC
Jean Luca Bez
Scientific Data Management, SciData Division, LBNL

NERSC Data Day 2022
October 26, 2022

I/O Stack: Moving Data To Disk

Productivity Interface is a thin layer on top of existing high performance I/O libraries, for productive big data analytics

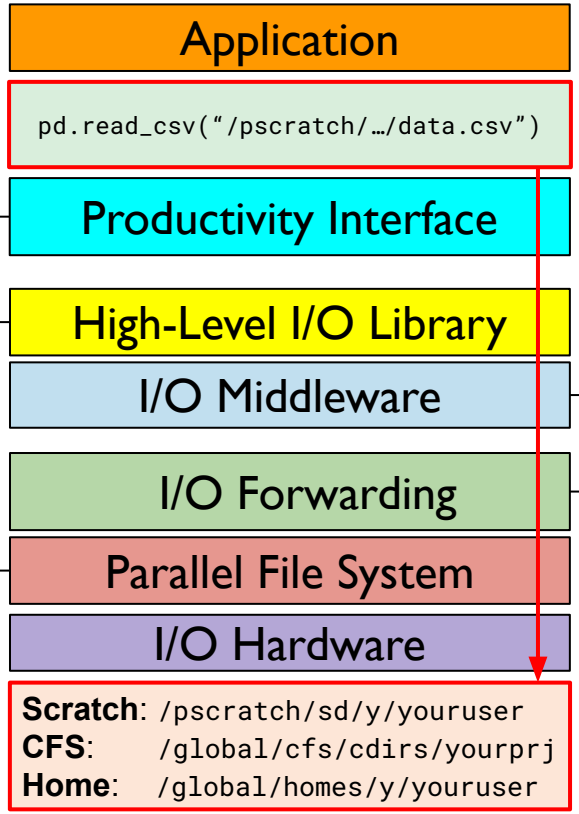
Python, Spark, TensorFlow

High Level I/O Libraries map application abstractions onto storage abstractions and provide data portability.

HDF5, Parallel netCDF, ADIOS

Parallel file systems maintain a logical file model and provide efficient access to data.

Lustre, GPFS, PVFS, PanFS



I/O Middleware organizes accesses from many processes, especially those using collective I/O.

MPI-IO, GLEAN, PLFS

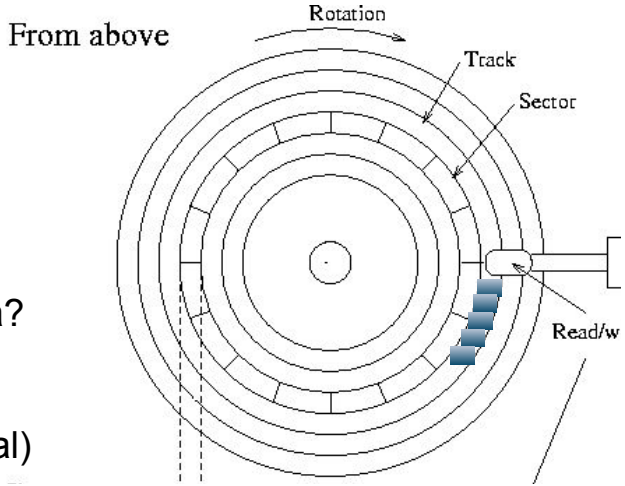
I/O Forwarding transforms I/O from many clients into fewer, larger requests; reduces lock contention; and bridges between the HPC system and external storage.

Cray DVS, IBM CIOD, IOFSL

I/O Pattern Analysis

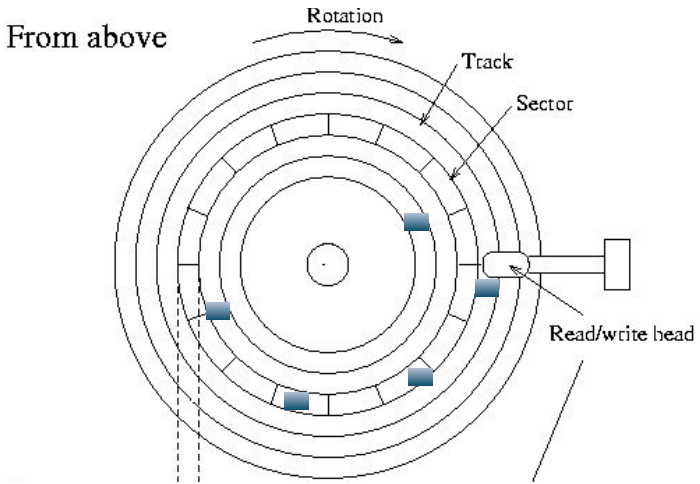
How to describe your I/O

- Number of Processes
- Number of Files
- Size per file
- Frequency of I/O
- Size per I/O
- Read, Write, Metadata?
- Shared File or not
- I/O Libraries
- Contiguous (Sequential) vs Non-contiguous (Random) access pattern
- Data alignment
- ...



Contiguous I/O

- Read time: 0.1ms



Noncontiguous I/O

- Seek time: 4ms
- Rotation time: 3ms
- Read time: 0.1 ms
- Total time = 7.1ms

I/O Profiling: Darshan

- Darshan
  - Lightweight HDF5/MPI-IO/POSIX I/O profiling tool, developed by ANL
    - https://www.mcs.anl.gov/research/projects/darshan/
  - Loaded by default at NERSC: currently version 3.4.0
    - https://docs.nersc.gov/tools/performance/darshan/

    - `module av -S darshan`

```
darshan/3.4.0 (D)    darshan/3.4.0-hdf5
darshan/3.3.1        darshan/3.3.1-hdf5
```

I/O Profiling: support for HDF5 et al.

- Darshan is available at NERSC with and without HDF5 support
- Darshan with HDF5 support
  - Makes all execs load HDF5 (latest `cray-hdf5-parallel`)
  - Dependency problems if you're using an old HDF5 (<1.10)
  - .. or when using different MPI libraries than Cray-MPI/MVAPICH
- You can build your own Darshan, see instructions in the docs https://docs.nersc.gov/tools/performance/darshan/#hdf5-aware-darshan-build
- parallel NetCDF profiling is also supported by Darshan, enable it at configure

I/O Profiling: Darshan

- Darshan log files produced at the end of successful executions of applications
  - All I/O calls are recorded, no sampling
    - I/O calls can spread several “layers”
    - i.e. NetCDF → HDF5 → MPI-IO → POSIX → Lustre (if I/O on scratch)
  - Log files can get very large depending on the number of process, I/O patterns used, etc

I/O Profiling: Darshan

- Darshan is a “post mortem” tool, no live profiling/debugging
  - Applications need to have darshan injected at compile time or manually loaded at runtime to profile I/O
- Cray Compiler wrappers (cc, CC, ftn) at NERSC inject Darshan into final exec
  - https://docs.nersc.gov/tools/performance/compilers/wrappers

```
$ cat hello.c; cc hello.c
int main() { return 0; }

$ ldd a.out |grep darshan
libdarshan.so => /path/…/lib/libdarshan.so
```

I/O Profiling: Darshan non-MPI

- Only MPI applications will trigger the tracing mechanism
  - Darshan overwrites `MPI_Init` and `MPI_Finalize`
  - Only applications that call `MPI_Finalize` will produce a darshan log file
  - For non-MPI applications, manually enable darshan with:

```
DARSHAN_ENABLE_NONMPI=1 \
LD_PRELOAD="$DARSHAN_BASE_DIR/lib/libdarshan.so" \
      your_application.py
```

  - Warning: do not export darshan in LD_PRELOAD or you'll trace any application, including `ls`
    - Impacts yours and other users' applications
  - MPI error with non-MPI exec? Build your own Darshan `--without-mpi`

I/O Profiling: Darshan log files

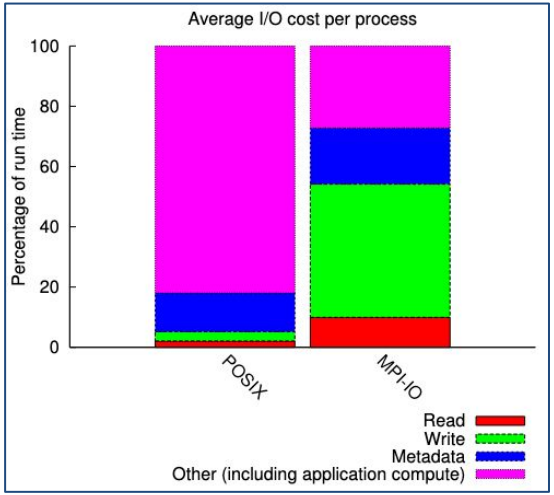
- Darshan log files
  - `/global/cscratch1/sd/darshanlogs/<year>/<month>/<day>/`
  - `/pscratch/darshanlogs/<year>/<month>/<day>/`
  - More than 1000 logs/day (more expected when Perlmutter system in production)
- Filename format:

`<username>_<jobname>_<jobid>_<time>_<date>-<uniqueid>-<timing>.darshan`

`elvis_vasp_id31418_231851_9-20-57716-7672284139862134 1237.darshan`

I/O Profiling: parsing Darshan logs

- Darshan scripts available at NERSC with any `darshan` module
  - `darshan-parser /input_file.darshan`
    - Parse content of Darshan log file and output text
    - Can be very verbose
  - `module load texlive`
    `darshan-job-summary.pl \`
    `/input_file.darshan`
    - Create a PDF report with useful I/O plots
    - More advanced plotting and analysis tools with DXT Explorer and Drishti



I/O Profiling: PyDarshan

- PyDarshan, new tool with Darshan 3.4.0
  - https://www.mcs.anl.gov/research/projects/darshan/docs/pydarshan/
  - ```
    $ pip install darshan
    $ python
    >>> report = darshan.DarshanReport('example.darshan')
    >>> report.records['STDIO'].to_df()
    ```
  - Version must match library available in `LD_LIBRARY_PATH`
  - Useful to process/analyze log files programmatically or build API interface
  - May segfault w/ Darshan <3.4.0 log files
    - Run `darshan-convert /example.darshan{,.converted}`
    ```


NERSC
Questions?