
DXT Explorer

Release 2.0

Jean Luca Bez, Hammad Ather, Suren Byna

Aug 08, 2023

GETTING STARTED

1	Build Instructions	3
1.1	Installing through git	3
1.2	Installing through pip	4
1.3	Build with Spack	4
1.4	Docker Image	4
2	Interactive Plots	7
3	Exploring	9
4	Operation Plot	11
5	Transfer Plot	13
6	Spatiality Plot	15
7	Rank Zero Heavy Workload Plot	17
8	Unbalanced Workload Plot	19
9	Stragglers Plot	21
10	I/O Phase Plot	23
11	OST Usage Operation Plot	25
12	OST Usage Transfer Plot	27
13	Ways to contribute	29
13.1	Reporting bugs	29
13.2	Suggesting enhancements	29
13.3	Testing	29
14	Citation	31
15	Copyright	33
16	License Agreement	35

DXT Explorer is an interactive web-based log analysis tool to visualize Darshan DXT logs and help understand the I/O behavior of applications. Our tool adds an interactive component to Darshan trace analysis that can aid researchers, developers, and end-users to visually inspect their applications' I/O behavior, zoom-in on areas of interest and have a clear picture of where is the I/O problem.

BUILD INSTRUCTIONS

DXT Explorer requires a Darshan log file collected with tracing data. The Darshan eXtended Tracing (DXT) support is disabled by default in Darshan. To enable tracing globally for all files, you need to set the `DXT_ENABLE_IO_TRACE` environment variable as follows:

```
export DXT_ENABLE_IO_TRACE=1
```

To enable tracing for particular files you can refer to the Darshan's documentation page.

1.1 Installing through git

Note: In Perlmutter (NERSC) you might need to load Darshan and Python modules if they are not already loaded. For other systems, please refer to their documentation to use the correct module name.

```
module load python
```

```
module load darshan
```

Note: In Summit at OLCF you need to follow this set of instructions.

```
module load python

conda create -n py310-dxt python=3.10
source activate py310-dxt
conda install arrow-cpp=10.0.1 pyarrow=10.0.1

git clone https://github.com/hpc-io/dxt-explorer
cd dxt-explorer

pip install .

dxt-explorer samples/YOUR-DARSHAN-FILE.darshan

conda deactivate
```

Run the below command to install some required Python libraries:

```
pip install -r requirements.txt
```

Then install dxt-explorer using the following command:

```
pip install .
```

1.2 Installing through pip

To install through pip, just run the following command:

```
pip install dxt-explorer
```

Warning: If you are installing dxt-explorer through pip, make sure the Darshan version installed on the machine matches the pyDarshan version installed through pip, otherwise you might get the following error:

```
darshan.discover_darshan.DarshanVersionError
```

Note: In NERSC systems (i.e., Cori or Perlmutter) you might need to load the Darshan module if it is not already loaded. For other systems, please refer to their documentation to use the correct module name.

```
module load darshan
```

1.3 Build with Spack

You can also use Spack to install dxt-explorer:

```
spack install dxt-explorer
```

Note: Use the following installation guide to install spack on your machine if it is not already installed: https://spack-tutorial.readthedocs.io/en/latest/tutorial_basics.html

1.4 Docker Image

You can also use a Docker image already pre-configured with all dependencies to run DXT Explorer:

```
docker pull hpcio/dxt-explorer
```

Since we need to provide an input file and access the generated .html files, make sure you are mounting your current directory in the container and removing the container after using it. You can pass the same arguments described above, after the container name (dxt-explorer).


```
docker run --rm --mount \
  type=bind,source="$(pwd)",target="/dxt-explorer" \
  dxt-explorer darshan/<FILE>.darshan
```


INTERACTIVE PLOTS

Interactive examples of DXT traces collected from FLASH, E2E, and OpenPMD are available in the [companion repository](#) of our PDSW'21 paper.

EXPLORING

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```

```
usage: dxt-explorer [-h] [-o OUTPUT] [-p PREFIX] [-t] [-s] [-i] [-oo] [-ot] [-r] [-u] [-st] [-d] [-l] [--start START] [--end END] [--from START_RANK] [--to END_RANK] [--browser] [-csv] [-v] darshan
```

DXT Explorer:

positional arguments:

darshan Input .darshan file

optional arguments:

-h, --help show this help message and exit
-o OUTPUT, --output OUTPUT Output directory
-p PREFIX, --prefix PREFIX Output directory
-t, --transfer Generate an interactive data transfer explorer
-s, --spatiality Generate an interactive spatiality explorer
-i, --io_phase Generate an interactive I/O phase explorer
-oo, --ost_usage_operation Generate an interactive OST usage operation explorer
-ot, --ost_usage_transfer Generate an interactive OST usage data transfer size explorer
-r, --rank_zero_workload Determine if rank 0 is doing more I/O than the rest of the workload
-u, --unbalanced_workload Determine which ranks have unbalanced workload
-st, --stragglers Determine the 5 percent slowest operations in the time distribution
-d, --debug Enable debug mode
-l, --list List all the files with trace
--start START Report starts from X seconds (e.g., 3.7) from beginning of the job
--end END Report ends at X seconds (e.g., 3.9) from beginning of the job
--from START_RANK Report start from rank N
--to END_RANK Report up to rank M

(continues on next page)

(continued from previous page)

--browser	Open the browser with the generated plot
--csv, --csv	Save the parsed DXT trace data into a csv
-v, --version	show program's version number and exit

DXT Explorer will generate by default an `index.html` file with links to all interactive plots that can be opened in any browser to explore. If the transfer or spatiality plots were enabled, additional `.html` files will be generated, one for each type and the link to those html files will be provided in the `index.html` file.



DARSHAN:

sample/jlbez_I0_test5.Linux_id1797024_12-1-78642-14680324743816948210_159.darshan

/gpfs/alpine/csc300/scratch/jlbez/vpic-brtnfld/results/vpic-hdf5-1662725/field_hdf5/T.1/fields_1.h5

OPERATION

TRANSFER

SPATIALITY

/gpfs/alpine/csc300/scratch/jlbez/vpic-brtnfld/results/vpic-hdf5-1662725/hydro_hdf5/T.1/hydro_proton_1.h5

OPERATION

TRANSFER

SPATIALITY

/gpfs/alpine/csc300/scratch/jlbez/vpic-brtnfld/results/vpic-hdf5-1662725/hydro_hdf5/T.1/hydro_electron_1.h5

OPERATION

TRANSFER

SPATIALITY

/gpfs/alpine/csc300/scratch/jlbez/vpic-brtnfld/results/vpic-hdf5-1662725/particle_hdf5/T.1/proton_1.h5

OPERATION

TRANSFER

SPATIALITY

/gpfs/alpine/csc300/scratch/jlbez/vpic-brtnfld/results/vpic-hdf5-1662725/particle_hdf5/T.1/electron_1.h5

OPERATION

TRANSFER

SPATIALITY

This is the expected console output when calling DXT Explorer:

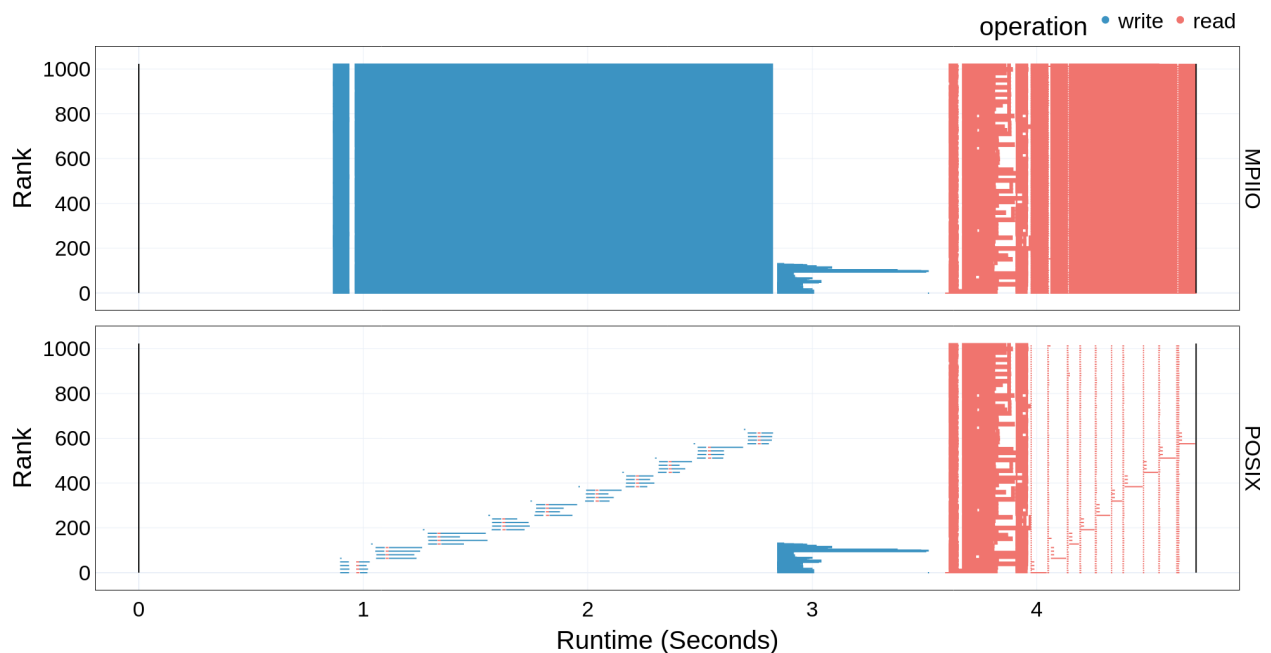
```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive operation for: <Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created Operation.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created Index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to
↪ interactively explore all plots
```

Interactive examples of DXT traces collected from FLASH, E2E, and OpenPMD are available in the [companion repository](#) of our PDSW'21 paper.

OPERATION PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the base `operation.html` plot. The `operation.html` plot shows the read and write operations performed by each rank throughout the runtime of the application. Contextual information link Rank, Operation, Duration, Size, Offset, Lustre OST can also be seen by hovering over a request.

This is the expected console output when calling DXT Explorer:

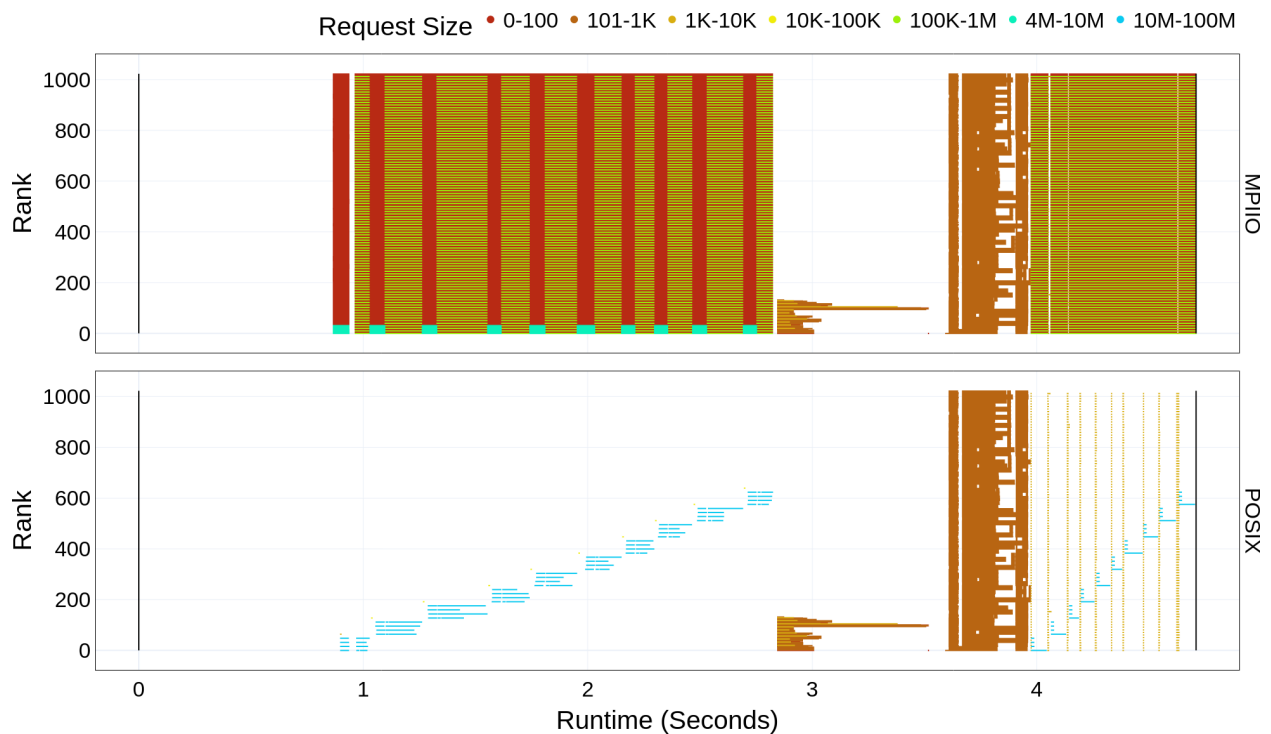
```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive operation for: <Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created operation.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to
↪ interactively explore all plots
```

Interactive examples of DXT traces collected from FLASH, E2E, and OpenPMD are available in the [companion repository](#) of our PDSW'21 paper.

TRANSFER PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer -t DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the `transfer.html` plot. The `transfer.html` plot shows the amount of data transferred by read and write requests of each rank throughout the runtime of the application. The request sizes are coloured in the plot based on the bin sizes used in Darshan, but the absolute value, if available, can be seen by hovering over a request. Contextual information link Rank, Operation, Duration, Size, Offset, Lustre OST can also be seen by hovering over a request.

This is the expected console output when calling DXT Explorer:

```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive transfer for: <Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created transfer.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
```

(continues on next page)

(continued from previous page)

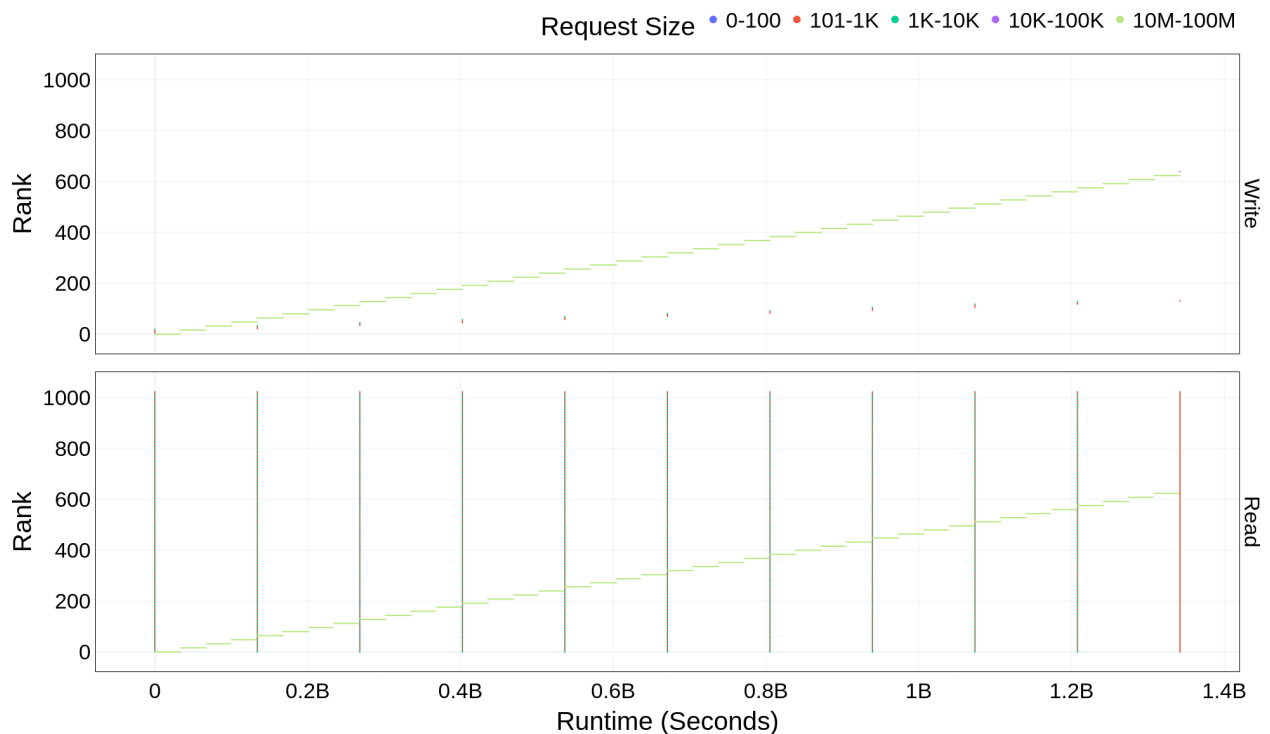
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to ↪
↪ interactively explore all plots

Interactive examples of DXT traces collected from FLASH, E2E, and OpenPMD are available in the [companion repository](#) of our PDSW'21 paper.

SPATIALITY PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer -s DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the base `spatiality.html` plot. Spatiality refers to the file offsets between consecutive I/O accesses. Typical spatial access patterns are contiguous, strided, or random. The `spatiality.html` plot shows the spatiality of the accesses in file made by each rank. Contextual information link Rank, Operation, Duration, Size, Offset, Lustre OST can also be seen by hovering over a request.

This is the expected console output when calling DXT Explorer:

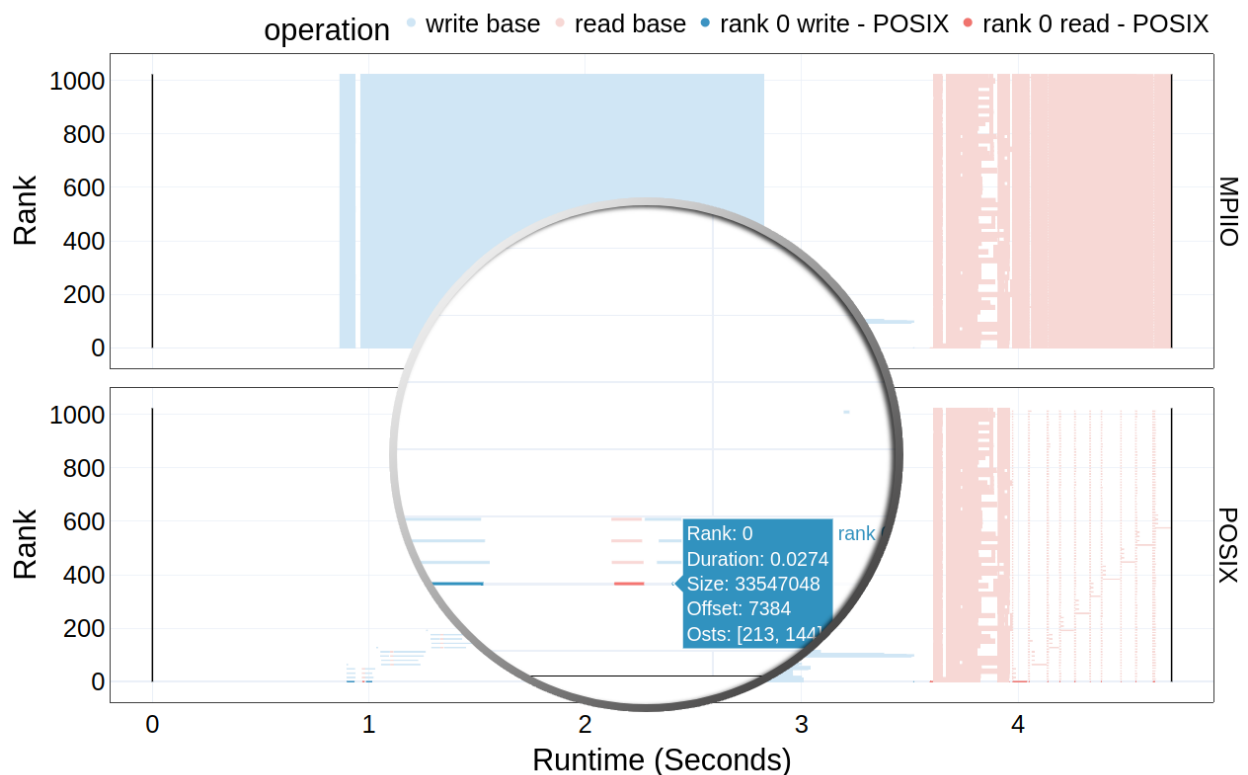
```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive spatiality for: <Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created spatiality.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to
↪ interactively explore all plots
```

Interactive examples of DXT traces collected from FLASH, E2E, and OpenPMD are available in the [companion repository](#) of our PDSW'21 paper.

RANK ZERO HEAVY WORKLOAD PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer -r DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the base `operation.html` plot. On the right of the `operation.html` plot, a dropdown menu will be displayed which will have an option to display rank zero heavy workload bottleneck, if that bottleneck exists. Upon clicking that button, the rank zero workload will be highlighted on the graph with the other operations in the background in an opaque color. Contextual information link Rank, Operation, Duration, Size, Offset, Lustre OST can also be seen by hovering over a request.

This is the expected console output when calling DXT Explorer:

```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive operation for: <Filename>
```

(continues on next page)

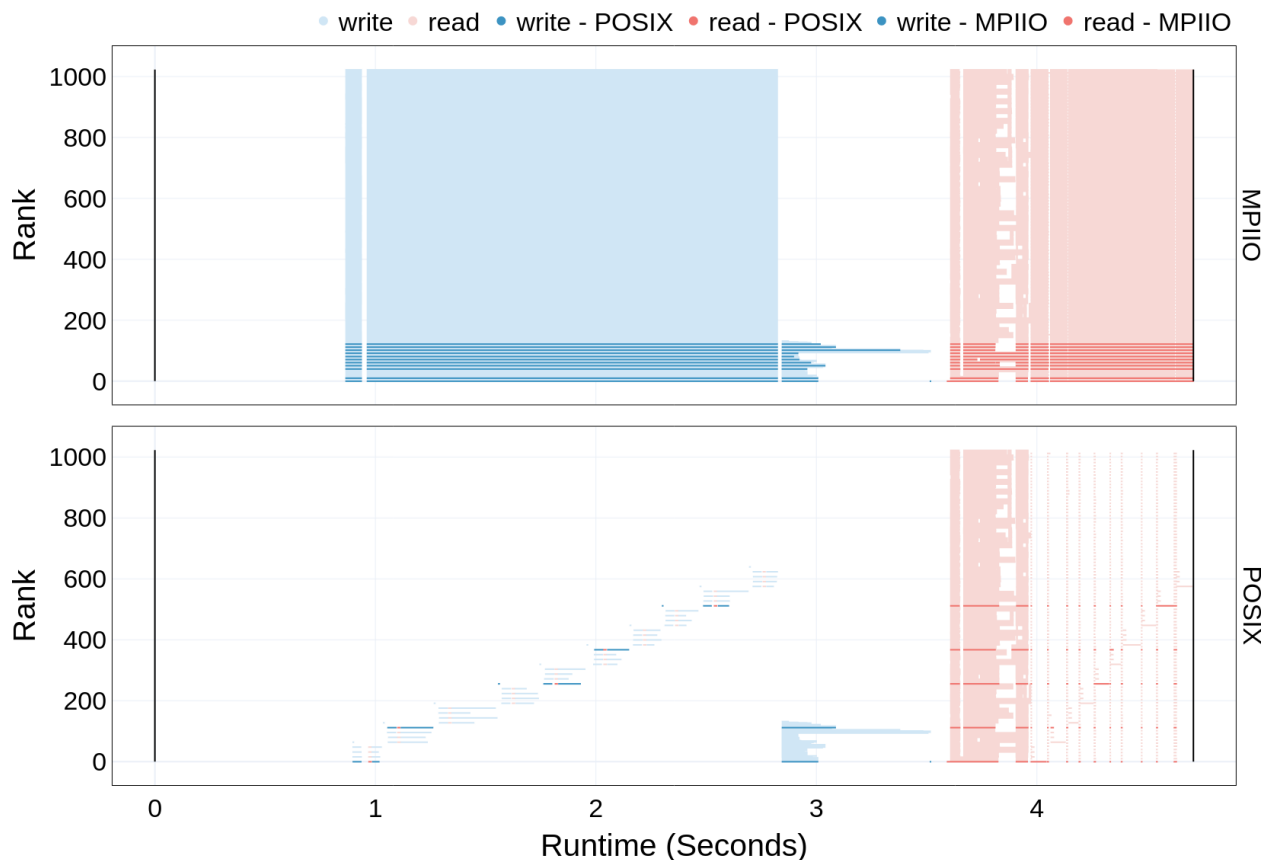
(continued from previous page)

```
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created operation.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to ↵
↵interactively explore all plots
```

UNBALANCED WORKLOAD PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer -u DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the base `operation.html` plot. On the right of the `operation.html` plot, a dropdown menu will be displayed which will have an option to display unbalanced workload, if that bottleneck exists. Upon clicking that button, the unbalanced ranks will be highlighted on the graph with the other operations in the background in an opaque color. Contextual information link Rank, Operation, Duration, Size, Offset, Lustre OST can also be seen by hovering over a request.

This is the expected console output when calling DXT Explorer:

```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
```

(continues on next page)

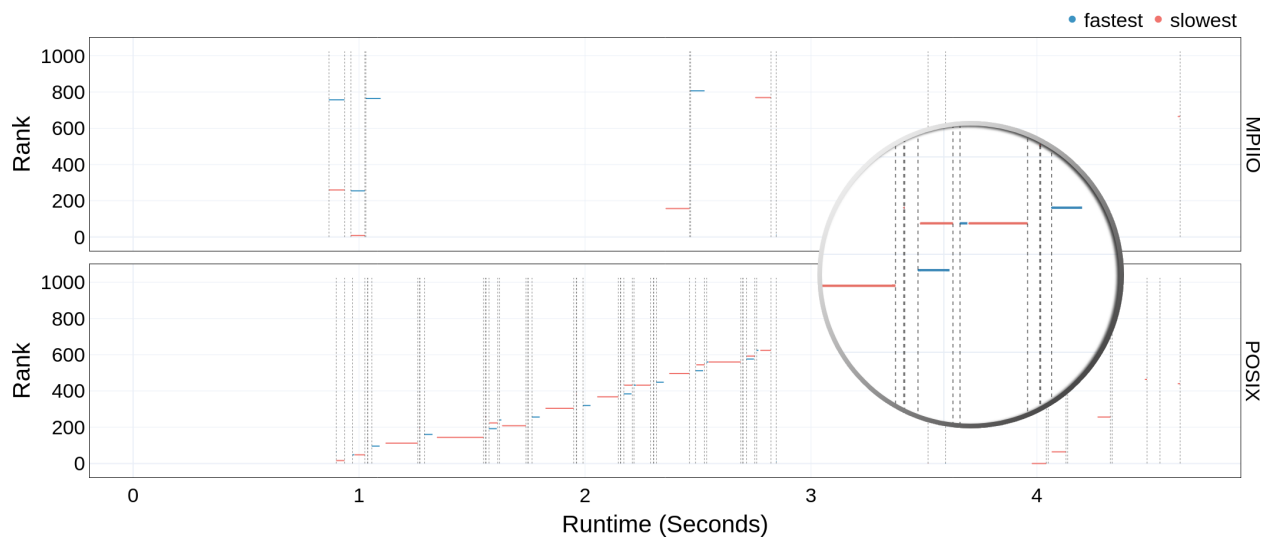
(continued from previous page)

```
2022-11-02 12:58:26,681 dxt - INFO - generating interactive operation for: <Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created operation.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to
↳interactively explore all plots
```


STRAGGLERS PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer -st DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the base `operation.html` plot. On the right of the `operation.html` plot, a dropdown menu will be displayed which will have an option to display stragglers, if that bottleneck exists. Upon clicking that button, the stragglers will be highlighted on the graph. Contextual information link Fastest Rank, Fastest Rank Duration, Slowest Rank, Slowest Rank Duration can also be seen by hovering over a request.

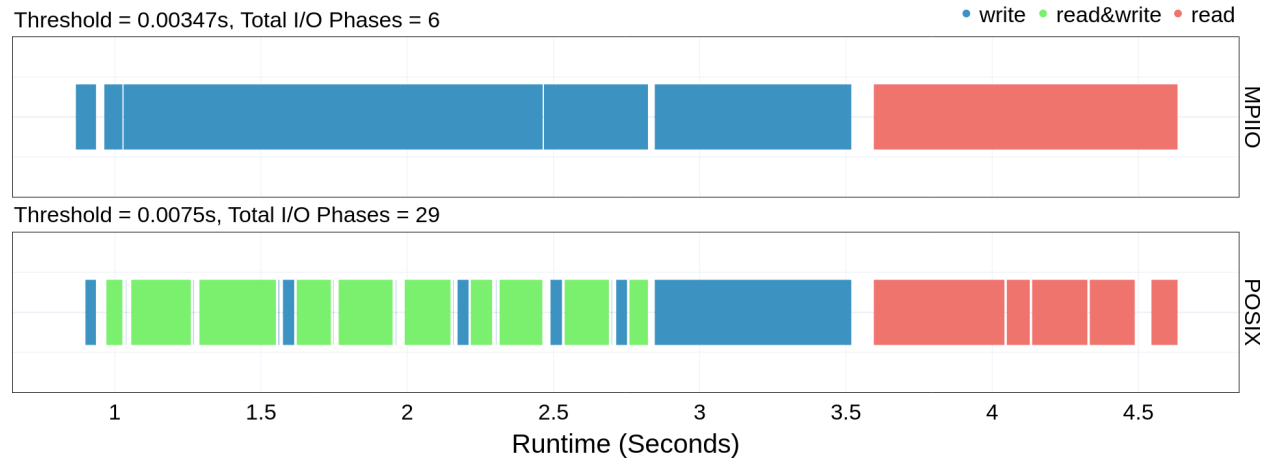
This is the expected console output when calling DXT Explorer:

```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive spatiality for: <Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created spatiality.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to ↵
↵ interactively explore all plots
```


I/O PHASE PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer -i DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the base `io-phase.html` plot. The `io-phase.html` plot shows the different I/O Phases in the data. The plot also shows information regarding the number of I/O phases in each interface (MPIIO and POSIX) and the threshold value used to merge the phases. The threshold value is computed by summing the mean and standard deviation of all the intervals between the I/O phases. Contextual information link `Fastest Rank`, `Fastest Rank Duration`, `Slowest Rank`, `Slowest Rank Duration` can also be seen by hovering over a phase.

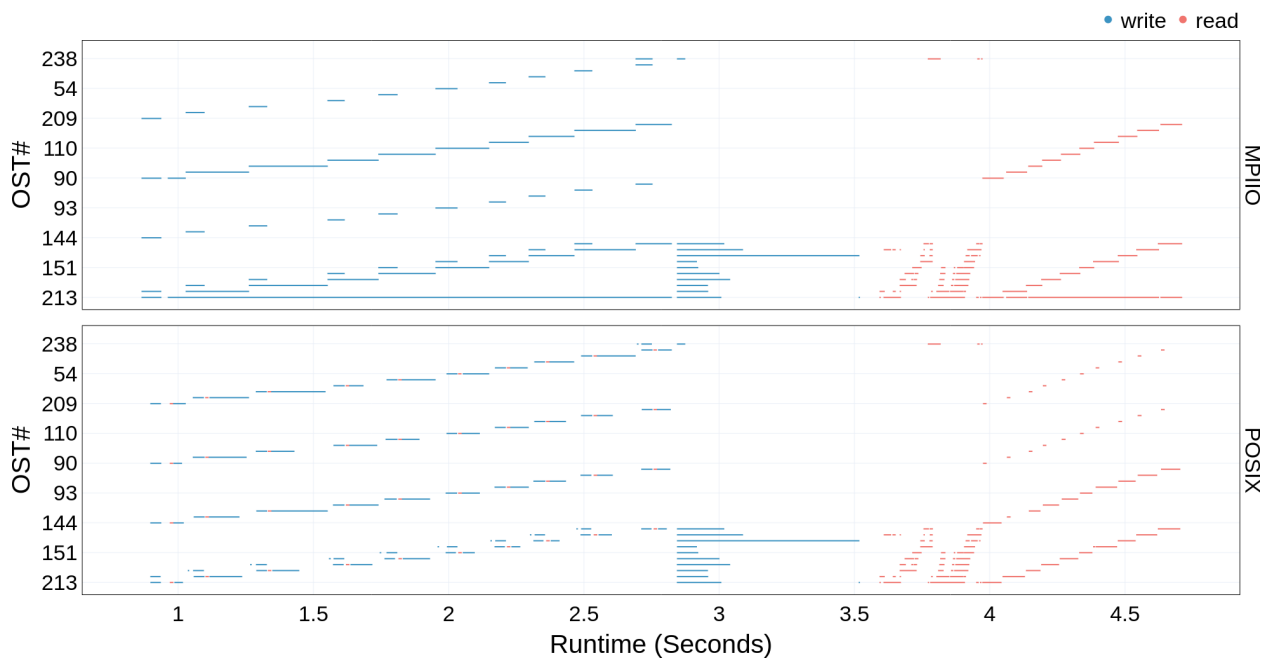
This is the expected console output when calling DXT Explorer:

```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive I/O phase plot for:
↳<Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created spatiality.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to
↳interactively explore all plots
```


OST USAGE OPERATION PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer -oo DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the `ost-usage-operation.html` plot. The `ost-usage-operation.html` plot shows the usage of each OST server for read and write operations throughout the runtime of the application.

Warning: This plot will only be generated if the application was executed on Lustre File System and Darshan collected those metrics.

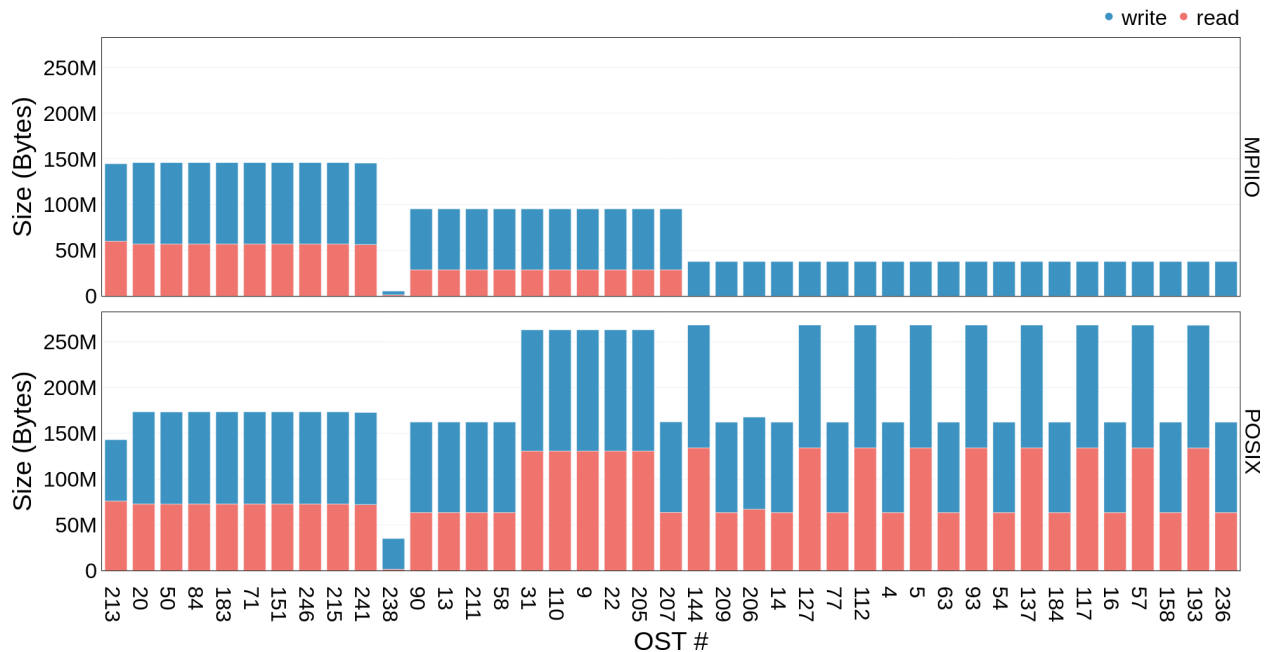
This is the expected console output when calling DXT Explorer:

```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive OST usage operation plot.
↳for: <Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created operation.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your browser to
↳interactively explore all plots
```


OST USAGE TRANSFER PLOT

Once the dependencies and DXT Explorer have been installed:

```
dxt-explorer -ot DARSHAN_FILE_COLLECTED_WITH_DXT_ENABLE.darshan
```



This will generate the `ost-usage-transfer.html` plot. The `ost-usage-transfer.html` plot shows the data transferred by each OST server for read and write operations throughout the runtime of the application.

Warning: This plot will only be generated if the application was executed on Lustre File System and Darshan collected those metrics.

This is the expected console output when calling DXT Explorer:

```
2022-11-02 12:58:22,979 dxt - INFO - FILE: <Filename> (ID <File ID>)
2022-11-02 12:58:22,979 dxt - INFO - generating dataframes
2022-11-02 12:58:26,681 dxt - INFO - generating interactive OST usage transfer plot for:
↳ <Filename>
2022-11-02 12:58:30,826 dxt - INFO - SUCCESS: <Path to the newly created operation.html>
2022-11-02 12:58:30,834 dxt - INFO - SUCCESS: <Path to the newly created index.html>
2022-11-02 12:58:30,834 dxt - INFO - You can open the index.html file in your
↳ browser to interactively explore all plots
```


WAYS TO CONTRIBUTE

We appreciate your interest in **DXT Explorer**, and thank you for taking the time to contribute!

We have compiled a set of instructions to help us make DXT Explorer even better.

13.1 Reporting bugs

You can open a new issue using our GitHub [issue tracker](#). If you run into an issue, please search first to ensure the issue has not been reported before. Open a new issue only if you have not found anything similar to your issue. Please, try to provide as much information as possible to reproduce your bug quickly.

13.2 Suggesting enhancements

You can use our GitHub [issue tracker](#) to describe your proposed feature. Please, provide the necessary context, covering why it is needed and what problem does it solve.

13.3 Testing

DXT Explorer constantly receives updates and improvements. If you can run the latest version, please consider helping us by reporting your findings, including bugs and performance regressions. Running DXT Explorer with different configurations and platforms helps us a lot in making it more robust by quickly identifying and solving issues.

CITATION

You can find more information about DXT Explorer in our PDSW'21 paper. If you use DXT in your experiments, please consider citing:

```
@inproceedings{dxt-explorer,  
  title = {{I/O Bottleneck Detection and Tuning: Connecting the Dots using Interactive_  
↵Log Analysis}},  
  author = {Bez, Jean Luca and Tang, Houjun and Xie, Bing and Williams-Young, David and_  
↵Latham, Rob and Ross, Rob and Oral, Sarp and Byna, Suren},  
  booktitle = {2021 IEEE/ACM Sixth International Parallel Data Systems Workshop (PDSW)},  
  year = {2021},  
  volume = {},  
  number = {},  
  pages = {15-22},  
  doi = {10.1109/PDSW54622.2021.000008}  
}
```


COPYRIGHT

DXT Explorer Copyright (c) 2022, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from the U.S. Dept. of Energy). All rights reserved.

If you have questions about your rights to use or distribute this software, please contact Berkeley Lab's Intellectual Property Office at IPO@lbl.gov.

Attention: This Software was developed under funding from the U.S. Department of Energy and the U.S. Government consequently retains certain rights. As such, the U.S. Government has been granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable, worldwide license in the Software to reproduce, distribute copies to the public, prepare derivative works, and perform publicly and display publicly, and to permit others to do so.

LICENSE AGREEMENT

DXT Explorer Copyright (c) 2022, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from the U.S. Dept. of Energy). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- (1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- (3) Neither the name of the University of California, Lawrence Berkeley National Laboratory, U.S. Dept. of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You are under no obligation whatsoever to provide any bug fixes, patches, or upgrades to the features, functionality or performance of the source code (“Enhancements”) to anyone; however, if you choose to make your Enhancements available either publicly, or directly to Lawrence Berkeley National Laboratory, without imposing a separate written license agreement for such Enhancements, then you hereby grant the following license: a non-exclusive, royalty-free perpetual license to install, use, modify, prepare derivative works, incorporate into other computer software, distribute, and sublicense such enhancements or derivative works thereof, in binary and source code form.