The I/O Trace Initiative: Building a Collaborative I/O Archive to Advance HPC

Nafiseh Moti
Johannes Gutenberg University Mainz
Mainz, Germany
moti@uni-mainz.de

André Brinkmann
Johannes Gutenberg University Mainz
Mainz, Germany
brinkman@uni-mainz.de

Marc-André Vef
Johannes Gutenberg University Mainz
Mainz, Germany
vef@uni-mainz.de

Philippe Deniel
CEA
Bruyères-le-Châtel, France
philippe.deniel@cea.fr

Jesus Carretero
Universidad Carlos III
Madrid, Spain
jcarrete@inf.uc3m.es

Jean-Thomas Acquaviva
DataDirect Networks
Paris, France
jtacquaviva@ddn.com

Philip Carns
Argonne National Laboratory
Illinois, USA
carns@mcs.anl.gov

Reza Salkhordeh
Johannes Gutenberg University Mainz
Mainz, Germany
rsalkhor@uni-mainz.de

ABSTRACT

HPC application developers and administrators need to understand the complex interplay between compute clusters and storage systems to make effective optimization decisions. Ad hoc investigations of this interplay based on isolated case studies can lead to conclusions that are incorrect or difficult to generalize. The I/O Trace Initiative aims to improve the scientific community's understanding of I/O operations by building a searchable collaborative archive of I/O traces from a wide range of applications and machines, with a focus on high-performance computing and scalable AI/ML. This initiative advances the accessibility of I/O trace data by enabling users to locate and compare traces based on user-specified criteria. It also provides a visual analytics platform for in-depth analysis, paving the way for the development of advanced performance optimization techniques. By acting as a hub for trace data, the initiative fosters collaborative research by encouraging data sharing and collective learning.

CCS CONCEPTS

• **Information systems → Hierarchical storage management**.

KEYWORDS

High Performance Computing, Storage systems, I/O profiling

ACM Reference Format:
Nafiseh Moti, André Brinkmann, Marc-André Vef, Philippe Deniel, Jesus Carretero, Jean-Thomas Acquaviva, Philip Carns, and Reza Salkhordeh. 2023. The I/O Trace Initiative: Building a Collaborative I/O Archive to Advance HPC. In *Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W 2023), November 12–17, 2023, Denver, CO, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3624062.3624192

SC-W 2023, November 12–17, 2023, Denver, CO, USA

ACM ISBN 979-8-4007-0785-8/23/11…$15.00
https://doi.org/10.1145/3624062.3624192

1 INTRODUCTION

The evolution of the high-performance computing (HPC) landscape has seen the integration of artificial intelligence/machine learning (AI/ML) workloads with traditional HPC applications. The different I/O behaviors of these applications have a significant impact on system performance [13, 18, 19], and the cost of I/O operations can dominate in large-scale simulations and AI/ML model training because of the widening gap between compute and storage performance. Yet, application developers often prioritize optimizing simulation code to save CPU cycles over I/O. Several reasons might explain this inclination: they believe that significant gains can be made with CPU optimization, perceive I/O inefficiencies as hardware limitations rather than software issues, or they might assume I/O is either already optimized or view its optimization as an external challenge beyond their immediate control or focus.

Entering this new era of computing therefore requires a thorough exploration of I/O patterns in order to optimize applications, user behavior, and storage systems. Unfortunately, it is not enough to explore selected applications with few changes in input parameters on a single HPC system, as doing so can easily lead to incorrect conclusions and optimizations. In contrast, it is necessary to evaluate many applications at different scales and inputs to ensure improvements in both application I/O efficiency and storage system efficiency.

This paper introduces the I/O Trace Initiative, which addresses these challenges by creating a comprehensive, open data archive of I/O traces. The initiative aims to provide a deeper understanding of I/O behavior across a wide range of scientific domains, applications, and compute clusters. The goal is to bridge the knowledge gap in understanding I/O behavior and provide valuable insights to system researchers and application developers alike. This initiative builds on and complements previous efforts in this area [3, 4, 6, 14, 17].

The I/O Trace Initiative differs from existing I/O trace efforts in that its first step is to collect as many different I/O traces as possible for a variety of major HPC applications across a variety of platforms. This collection will provide crucial context that enables software architects to understand whether their applications are sensitive to unexpected input parameters or HPC machine configurations and to identify scaling limits. In addition, HPC storage administrators will be able to compare the execution behavior of applications running on their HPC platform with the I/O behavior across different platforms, allowing them to find and adjust weaknesses in their storage configuration.

The I/O archive and its associated website go far beyond a simple data repository by providing visualization tools for analyzing individual trace files, comparing different trace files, and merging traces of (non-MPI) workflows into a single, unified representation. It thus provides the basis for collaborative learning in the field of HPC tracing. It is based on the trace format introduced by Darshan [6] and uses and extends many of Darshan's analysis capabilities.

However, the development of such an archive poses several challenges, including data privacy, data uniformity, and interpretation of results. In addition, contributors to the archive must be motivated to spend time tracing and submitting traces, for example by being able to cite their work. These challenges are addressed in the paper, along with our solutions in the design and implementation of our system. Thus, this paper highlights the importance of building an open access archive of I/O traces from HPC workloads and presents our initiative as a response. The challenges and potential of this initiative are discussed in detail.

This paper is organized as follows: Section 2 explores the motivation and state of the art in I/O profiling, emphasizing the need for an open access trace archive. It then outlines challenges and benefits of our initiative for designers, researchers, and developers. The framework section details our archive's design and potential through examples, aiming to enable I/O studies for HPC and AI/ML applications and foster collaboration and open science principles. Section 4 concludes and highlights future research directions.

2 BACKGROUND AND MOTIVATION

HPC and AI/ML workloads exhibit a wide range of I/O behaviors due to varying computational needs and data dependencies. The nature of data models, access patterns, and I/O operations, including both POSIX functions, and more complex MPI-IO operations, are determinants of the application's overall performance [2]. Profiling tools like Tau [20], Score-p [12], IOPin [11], Recorder [15, 22], and Darshan [5, 21] are essential, providing detailed data and insight into the interplay of I/O, computation, and communication in overall execution time.

2.1 I/O Profiling with Darshan

Darshan is an I/O profiling tool widely adopted in the HPC community because of its lightweight design and focus on high-level application behavior. It provides valuable insights into the interaction between applications and the I/O subsystem, offering metrics such as I/O operation volume, time spent in I/O, and performance information for different layers of the I/O stack.

The I/O Trace Initiative has selected Darshan as its foundation for collecting traces for several reasons. Its application-centric perspective offers high-level behavioral insights rather than low-level system metrics, providing a more holistic understanding of the application I/O. Additionally, its scalable and portable nature ensures compatibility with diverse HPC environments, enabling us to capture a broad range of data. Moreover, the new Darshan eXtended Tracing (DXT) module [25] allows varying the level of fidelity captured by Darshan at run time without modifying or recompiling applications. Darshan's profiling capabilities allow it to profile the various layers of the I/O stack, thus providing detailed I/O statistics that facilitate more nuanced analyses. Furthermore, the community around Darshan constantly enhances and extends the tool, assuring ongoing support that will greatly benefit our archive project.

Despite the merits of profiling tools like Darshan, however, they do not fully capture the application context and thus lead to limitations in performance optimization. Notably, certain metadata about applications, workflows, and varying workloads, especially in non-MPI applications, may not be preserved, resulting in a knowledge gap. Furthermore, the use of profiling tools in isolation lacks a structured mechanism for comparing and contrasting I/O behaviors across diverse applications, systems, and scales.

As a solution, an I/O trace archive is deemed necessary to encapsulate the application context and facilitate meaningful insights into I/O behavior under various scenarios. The proposed archive can facilitate bridging the gap between I/O profiling and acquiring essential information, while also handling the challenges stemming from system heterogeneity. Additionally, fostering close collaboration with the scientific community plays a vital role for Darshan developers in detecting bugs and analyzing application requirements, further emphasizing the significance of the proposed I/O trace archive.

2.2 Challenges and Motivations for System, Storage, and Application Researchers

For system, storage, and application researchers, the complexities associated with tracing and analyzing I/O operations present multifaceted challenges. Current tools for I/O profiling tend to offer a limited scope, often lacking in the capacity to provide applications' context and metadata. Crucial information, such as system configurations, software versions, and specifics of the application's use case, processed data types, and algorithms used, is often not documented, hindering a full-fledged analysis.

Simultaneously, the need for longitudinal tracking of I/O performance, currently lacking in most profiling tools, is growing. Such functionality is invaluable for understanding the effectiveness of system changes and optimization techniques over time. Furthermore, with the increasing prevalence of AI/ML workloads in HPC systems, there is a pressing need to explore and understand I/O characteristics specific to widely used AI frameworks such as TensorFlow and PyTorch, an aspect currently not adequately covered.

Additionally, researchers and developers across different fields of science often work in isolation, with limited awareness of similar system and I/O requirements in other disciplines. This compartmentalization can lead to redundancies and prevent the cross-fertilization of ideas and methodologies.

The motivation for the I/O Trace Initiative is born out of these unmet needs. By providing a rich, contextually aware, and comprehensive dataset of I/O traces, we aim to cater to the needs of system, storage, and application researchers alike. This archive is expected to enhance the understanding of I/O behaviors and their dependencies on system and application context, impacting overall application performance.

The initiative is also committed to empowering application developers. The archive's access to detailed, context-rich information and support for comparative and longitudinal analysis will guide developers to better comprehend their application's I/O behaviors and optimize performance. Furthermore, it seeks to contribute to the AI/ML development community by enhancing the understanding of I/O patterns specific to AI/ML workloads. By encouraging cross-disciplinary collaboration, we believe this initiative will stimulate innovation and efficiency across scientific disciplines, benefiting the wider research community.

2.3 Related Work

Several studies have utilized Darshan traces for various aspects of I/O behavior in HPC systems. For example, Khetawat et al. used augmented Darshan traces with the I/O replay and CODE simulation suite to focus on burst buffers [10].

Others explored I/O load and behavior in AI/ML applications using Darshan traces. Chien et al. introduced tf-Darshan, connected with the TensorFlow Profiler, for fine-grained I/O analysis in machine learning [7]. Recent works such as [18] generated HPC traces assited by a machine learning pipeline, and [2] provided baseline taxonomy for I/O patterns as a common ground for HPC I/O researchers.

Some tools extend Darshan for interactive trace data analysis. Examples include DXT Explorer for detecting I/O bottlenecks [3], coupling Darshan with TOKIO to tune HPC workflows [14], and IOMiner for analyzing heterogeneous I/O logs [23]. Unlike the I/O Trace Initiative, these approaches do not aim for a public and interactive I/O trace repository. However, their insights and methodologies will influence the functionality of the HPC I/O tracing archive.

Several analysis tools were evaluated using ALCF I/O data repository traces [4]; however, these were limited to a single (now decommissioned) platform with anonymized logs and did not provide an interactive interface for exploring them. Our I/O Trace Initiative complements such efforts, offering an open repository of I/O traces with context and metadata. It can enhance workload understanding and aid the HPC community in I/O research and performance tuning.

2.4 FAIR Principles

Our work is fundamentally guided by the FAIR Principles, standing for *Findability, Accessibility, Interoperability,* and *Reusability*, introduced by Wilkinson et al. [24]. These principles are globally accepted standards for managing digital resources efficiently.

Findability prioritizes the discoverability of digital assets. We integrate this principle by assigning a permanent DOI to each I/O trace and allowing detailed metadata tagging, thus enhancing searchability in our I/O Trace Initiative.

Accessibility advocates for straightforward access to resources post-discovery. We support this by providing an open-access platform and lucid data access protocols for our I/O traces.

Interoperability necessitates compatibility of data with other datasets. We achieve this by utilizing Darshan, a widely accepted I/O profiling tool, which produces traces in an interoperable format.

Reusability emphasizes that data should be sharable for future research. Our initiative ensures this by providing comprehensive metadata for each trace and using a data format familiar to the HPC community. Hence, our approach encapsulates the FAIR principles, optimizing the usability and effectiveness of our I/O profiling efforts.

3 THE I/O TRACE INITIATIVE FRAMEWORK

The I/O Trace Initiative provides a complete framework for collecting, managing, studying, and interpreting I/O traces. Based on the principles of open science, this project encourages data sharing and interdisciplinary collaboration. It has been designed to meet the practical needs of researchers, as evidenced by feedback from participants in several European projects. The specific features and design considerations of our project are discussed in detail in the following sections.

3.1 The Central Repository: A Nucleus for I/O Trace Investigation

Acting as the central repository, the I/O Trace Initiative website streamlines the process of submitting and exploring I/O traces. Driven by a Go backend and an interactive frontend managed by *React*, the website is highly capable. The backend can process Darshan log files using both the `darshan log-utils` C API [1] and `PyDarshan`, although PyDarshan is preferred because it offers a more comprehensive collection of reusable analysis routines. A no-SQL database was implemented to allow easy expansion of stored data and the inclusion of additional information as required. This also enables the storage of log data created by different tracing applications in the future without the need to match them individually with data stored by Darshan. The frontend offers interactive charts that can be filtered by different I/O sources and types.

The website provides a variety of interactive data visualizations, such as heatmaps of I/O intensity, bar plots of operations, histograms of I/O sizes, and cumulative Pareto plots of I/O volume per process. The website serves as a platform for cross-collaboration, enabling researchers and developers to contribute, observe, analyze, and compare traces across different scientific domains.

In order to accommodate the common need among researchers to run customized analysis on the data, an API based on the *OpenAPI* standard [16] is provided for retrieving data from the backend. Users can employ automated code generation tools to access the API and receive the results in a structured manner. All APIs used by the frontend will also be made available to the public.

The subsequent sections will delve into the specific features and design considerations of the website.

3.2 The Submission Interface

Our website is equipped with a submission form that facilitates the uploading of Darshan profiles by collaborators. As part of this process, users provide certain metadata about the profile, including details such as their name, affiliation, the name of the cluster on which the application ran, and the name of the application itself.

This wealth of metadata preserves the application context, allowing users to grasp the distinctive behaviors exhibited by diverse workloads. Providing metadata enables others to more easily find relevant traces and compare them with greater confidence. However, we do not strictly require users to provide all the available information.

Anonymization. Darshan log files contain information that users might consider sensitive. For instance, the paths of all accessed files are stored in the log file. Darshan provides an obfuscation functionality that randomizes the directory path and file names in the trace file. We apply this obfuscation to the uploaded traces to remove the sensitive data.

Privacy policy. On our submission site, we collect names and email addresses of authors and correspondents. This information is required to authenticate and verify the submitted I/O profiles and associated metadata, maintaining its integrity within our public archive. Additionally this information allows for necessary exchanges with the authors during the verification process, aiding in the assurance of data accuracy and reliability in the HPC and storage domains. After data verification we delete all personal data in line with data protection regulations, unless the data contributors request their names to be linked to the data via a Digital Object Identifier (DOI). Such requests result in the permanent storage of the DOI, associating the contributors' names with the data in open, external databases. This approach is consistent with our policies that advocate for transparency and accuracy in data processing and storage. It also reflects our commitment to preserving the data we manage.

3.3 Archive and Search Capabilities

Following verification, I/O traces are cataloged in a searchable archive created using Elasticsearch [9] to facilitate rapid and efficient search capabilities. This archive is instrumental in enabling users to navigate, filter, and compare various traces, acting as an essential tool in understanding performance fluctuations across different systems and configurations.

Despite the robust search functionalities, we note that because of the anonymization process applied to maintain user privacy, where directory paths and file names in the trace file are randomized, search at the record level based on file names of operations and similar criteria, is not supported. Consequently, the system is unable to categorize files in a manner that preserves information about the type or purpose of the file post-anonymization.

Nonetheless, the website provides a comparison table and search functionality allowing users to filter and compare traces based on a variety of different metrics such as execution time, number of workers/ranks, recognizable workload characteristics like checkpointing, or application names, thus retaining a rich landscape of options for comparison and analysis.

3.4 In-Depth Analysis and Visualization Tools

The I/O Trace Initiative platform serves not only as a data repository, but also as a comprehensive analytics tool. For each workload uploaded, the platform generates a detailed analysis accompanied by a series of visualizations that give insights into I/O patterns and behaviors.

Workload details. For every uploaded trace file, the platform displays the information provided by the uploader about the application and its running environment. Additionally, it offers insights into the overall count of different I/O operations, giving users an overview of the workload details.

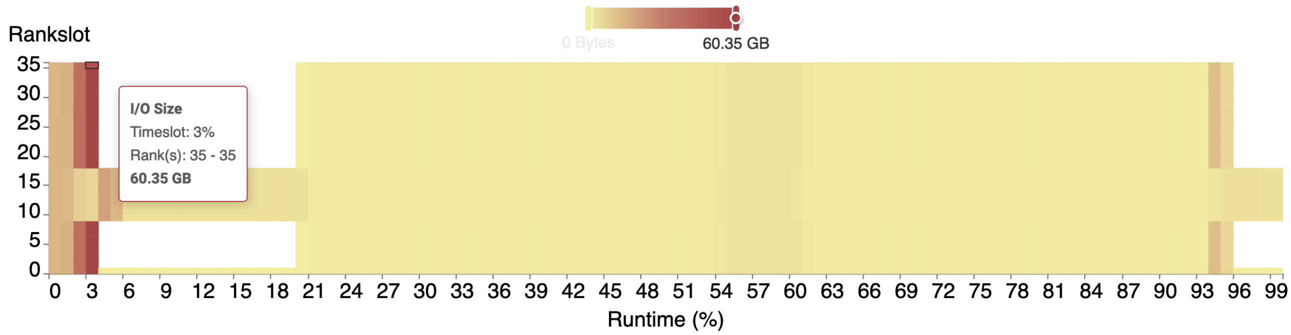
Heatmap visualization. The platform presents heatmaps that depict the I/O activity per timeslot and per rank (see Figure 1). HPC applications often have long runtimes, which implies that a trace can contain several hundreds or even thousands of timeslots. With the number of ranks also potentially reaching thousands, a heatmap can represent millions of data points. Full detailed visualization of such extensive data can be challenging. Therefore, to provide a high-level overview of the I/O activity and allow users to discern the I/O pattern of an HPC workload swiftly, we implement a scaling mechanism for both timeslots and ranks. We reduce the timeslots to a fixed number, 100, each representing 1% of the total execution time. This enables users to compare applications with different runtimes.

To scale down ranks, we group them into 200 bins, enabling users to perceive the overall I/O activity flow on small screens and easily access each data point's details. We store full-resolution data, accessible by downloading the original Darshan log file or via our API providing full-resolution heatmap data.

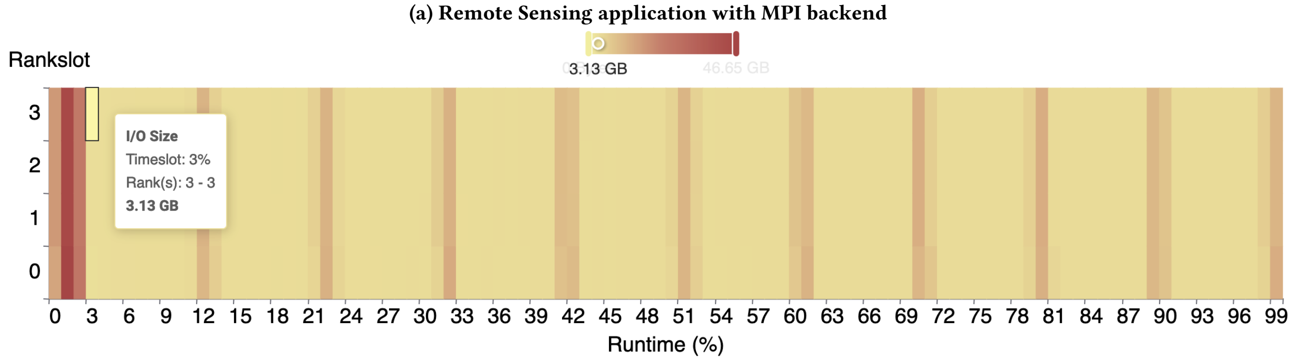
According to our observations from the collected traces, the logarithmic scale used in Darshan summary generation is not universally suitable for all workload types. Therefore, we provide users with an option to switch between normal or logarithmic scales for heatmaps. Furthermore, users can filter their interested range to find the corresponding timeslots and ranks.

Comparison tools. One of the central aims of this platform is to facilitate the comparison of various traces, allowing users to identify commonalities and differences among their target applications and configurations. Consequently, users can select multiple traces and observe their shared properties and disparities. They can then refer to a figure that shows the I/O volume between these traces over their runtime. The platform normalizes runtimes into 100 timeslots, making it possible to compare different applications with varying runtimes. This normalization also enables the comparison of different traces of the same application with different problem sizes and/or number of ranks. The scaling of I/O volume relative to the problem size can provide crucial insights into application scalability and potential I/O bottlenecks, benefiting both application developers and users.

Pareto CDF chart. A Pareto chart combines the graphical properties of a bar chart and a line chart and is based on the Pareto principle. This principle states that approximately 80% of the effects are caused by 20% of the causes. Using this concept, a Pareto chart can be instrumental in highlighting dominant factors in a dataset and, in our case, identifying collective or localized I/O patterns. Our website provides cumulative Pareto charts that allow users to easily observe the distribution of I/O accesses. It also helps identify the ranks that are most affected by I/O performance.

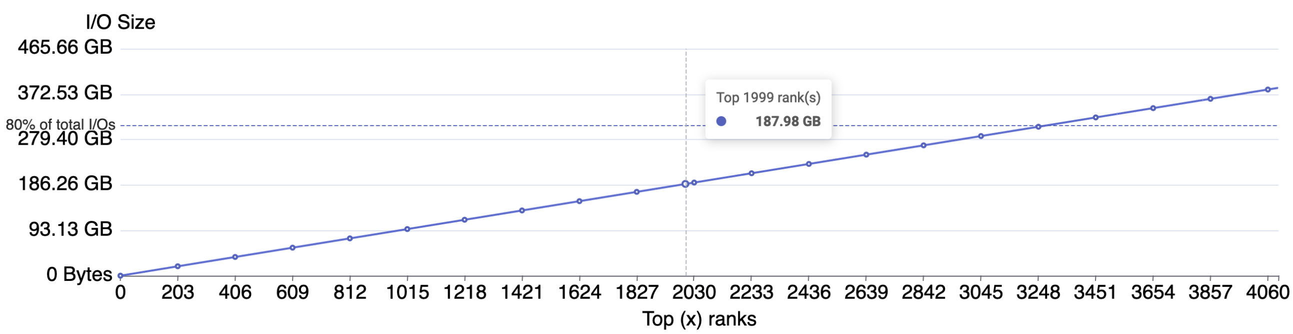


(a) Remote Sensing application with MPI backend

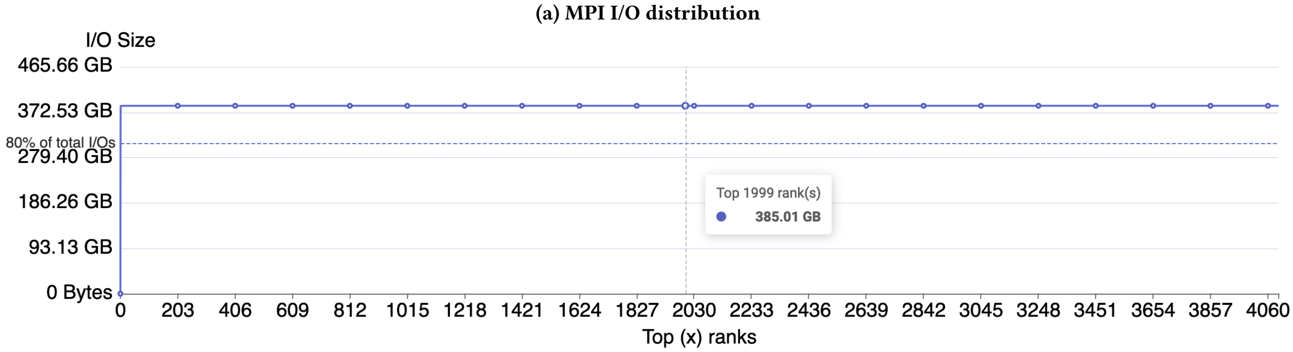


(b) Remote Sensing application with Gloo backend

Figure 1: Comparative Heatmaps of MPI and Gloo backends for Satellite Classification Workload



(a) MPI I/O distribution



(b) POSIX+STDIO I/O distribution

Figure 2: I/O volume distribution

To construct a cumulative Pareto chart, we order the I/O contributions of different ranks from highest to lowest based on their respective I/O sizes (see Figure 2). The cumulative I/O frequency is then graphically represented.

Pareto analysis is performed separately for each I/O type, allowing application developers and HPC administrators to identify potential read and write path optimizations. Figures 2a and 2b illustrate insights from the cumulative plot for the turbulent flow simulation with 4,096 ranks using Nek5000[1]. For example, the MPI-IO pattern shows a uniform distribution, while POSIX+STDIO shows that almost all I/O is performed by a single rank. The total I/O per rank is visualized in another graph.

Non-MPI workloads. Not all applications running in HPC environments use MPI. For example, many AI workloads manage their processes internally. However, Darshan can only aggregate the I/O accesses of MPI applications and must generate a per-process trace file for non-MPI workloads. This factor makes it difficult to analyze AI applications using standard Darshan parsing tools.

To overcome this challenge, our platform allows users to upload a compressed file, containing multiple Darshan log files. Each file is processed individually, and then all subresults are combined. Since Darshan log files do not provide a correlation between the processes, we sort the files alphabetically and assign rank numbers accordingly.

Darshan records the I/O for these processes individually and does not synchronize the timeslots used for heatmaps. This discrepancy can lead to variations in the starting time of different non-MPI processes. When combined, such variations can result in inaccuracies in the overall heatmap generated by our platform. Aligning the timeslots is a nontrivial task that could introduce more inaccuracies. Thus, we display the combined heatmap and include a disclaimer indicating the potential inaccuracy of the information therein.

Figure 1a and Figure 1b present heatmaps generated from the same classification workload dataset derived from the Sentinel-2 satellite's remote sensing data. The classification workload was executed on the same training data to ensure a direct comparison between the two different configurations of the Horovod framework. Specifically, Figure 1a illustrates the I/O behavior of a configuration utilizing the MPI backend with 36 ranks, while Figure 1b demonstrates the performance of the Gloo backend using 4 workers on 4 GPUs. Note that in the Gloo configuration, each worker generates a separate Darshan log file and all log files are then merged into a single visualization on our platform. This approach allows for an in-depth comparison of the I/O behavior and of the computational efficiency across different backend configurations and their respective handling of satellite data analysis workloads.

3.5 Longitudinal Data Management and Citability

To ensure data longevity and to facilitate citation, the I/O Trace Initiative adheres to best practices in scientific data management, all integrated into our user-friendly website workflow. When a submitter uploads I/O traces and associated metadata through the I/O Trace Initiative website, the platform interacts seamlessly with the *Zenodo* repository. Zenodo is a general-purpose open access repository developed under the European OpenAIRE program and hosted by CERN [8]. This integration allows the permanent deposit of traces and associated metadata in a recognized repository.

Immediately after the deposition process, Zenodo automatically generates and assigns a unique DOI to each dataset. This DOI serves as a permanent identifier that makes the data easily citable in academic literature, and it is promptly relayed back to the submitter via the I/O Trace Initiative website.

This practice not only gives contributors credit for their valuable data but also enhances the transparency and reproducibility of HPC I/O research. The integrated web-based workflow assures contributors that their submission will be preserved for future use, cited appropriately when used by others, and made available to the broader scientific community to facilitate open and collaborative research.

4 CONCLUSION

As the volume and complexity of data in high-performance computing and artificial intelligence/machine learning applications increase, effective I/O management becomes critical for achieving efficient system performance. Our I/O Trace Initiative is a significant step toward this goal, providing a versatile platform for I/O trace collection, in-depth analysis, and cross-discipline collaboration.

Accessible via https://hpcioanalysis.zdv.uni-mainz.de, this initiative serves as a hub where researchers and professionals can find approved traces and collaborate by uploading their own traces for community access. The platform offers a range of comprehensive features, including the submission of Darshan profiles, as well as archive and search functionalities. Coupled with detailed visualization tools, these features facilitate a deeper understanding of I/O patterns and behaviors across a broad spectrum of applications.

By adhering to the FAIR principles and promoting open science, we aim to make I/O traces and their analyses more accessible, fostering transparency, reproducibility, and collaboration. The I/O Trace Initiative is not just a tool but a vision for a more collaborative, insightful future in HPC and AI/ML research. As our platform evolves, we remain dedicated to serving the HPC and AI/ML communities, empowering them to enhance I/O performance optimization.

ACKNOWLEDGMENTS

This work was partially funded by the European Union's Horizon 2020 and the German Federal Ministry of Education and Research (BMBF) under the EuroHPC Joint Undertaking (JU) ADMIRE (GA No. 956748, BMBF funding No. 16HPC007) and IO-SEA (GA No. 955811, BMBF funding No. 16HPC027). Further funding was provided by the "National Research Data Infrastructure for and with Computer Science (NFDIxCS)" project, which is funded by German Research Foundation (DFG) with project number 501930651. Lastly, this research also received support from the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357.

[1]https://nek5000.mcs.anl.gov

REFERENCES

[1] 2015. darshan-logutils.c. https://github.com/darshan-hpc/darshan/blob/main/darshan-util/darshan-logutils.c

[2] Jean Luca Bez, Suren Byna, and Shadi Ibrahim. 2023. I/O Access Patterns in HPC Applications: A 360-Degree Survey. *ACM Comput. Surv.* (jul 2023).

[3] Jean Luca Bez, Houjun Tang, Bing Xie, David B. Williams-Young, Robert Latham, Robert B. Ross, Sarp Oral, and Suren Byna. 2021. I/O Bottleneck Detection and Tuning: Connecting the Dots using Interactive Log Analysis. In *6th IEEE/ACM International Parallel Data Systems Workshop (PDSW@SC), St. Louis, MO, USA, November 15.* 15–22.

[4] Philip Carns. 2013. *ALCF I/O Data Repository.* Technical Report. Argonne National Lab.(ANL), Argonne, IL (United States).

[5] Philip Carns. 2014. Darshan. In *High performance parallel I/O.* Chapman and Hall/CRC, 351–358.

[6] Philip H. Carns, Kevin Harms, William E. Allcock, Charles Bacon, Samuel Lang, Robert Latham, and Robert B. Ross. 2011. Understanding and Improving Computational Science Storage Access through Continuous Characterization. *ACM Trans. Storage* 7, 3 (2011), 8:1–8:26.

[7] Steven WD Chien, Artur Podobas, Ivy B Peng, and Stefano Markidis. 2020. tf-Darshan: Understanding fine-grained I/O performance in machine learning workloads. In *2020 IEEE International Conference on Cluster Computing (CLUSTER).* IEEE, 359–370.

[8] European Organization For Nuclear Research and OpenAIRE. 2013. Zenodo. https://doi.org/10.25495/7GXK-RD71

[9] Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine.* " O'Reilly Media, Inc.".

[10] Harsh Khetawat, Christopher Zimmer, Frank Mueller, Sudharshan Vazhkudai, and Scott Atchley. 2018. Using darshan and codes to evaluate application i/o performance. *SC Poster Session* (2018).

[11] Seong Jo Kim, Seung Woo Son, Wei-keng Liao, Mahmut T. Kandemir, Rajeev Thakur, and Alok N. Choudhary. 2012. IOPin: Runtime Profiling of Parallel I/O in HPC Systems. In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis, Salt Lake City, UT, USA, November 10-16, 2012.* IEEE Computer Society, 18–23.

[12] Andreas Knupfer, Christian Rossel, Dieter an Mey, Scott Biersdorff, Kai Diethelm, Dominic Eschweiler, Markus Geimer, Michael Gerndt, Daniel Lorenz, Allen Malony, and Wolfgang E. Nagel. 2012. Score-P: A Joint Performance Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir. (8 2012). https://www.osti.gov/biblio/1567522

[13] Thorsten Kurth, Sean Treichler, Joshua Romero, Mayur Mudigonda, Nathan Luehr, Everett H. Phillips, Ankur Mahesh, Michael A. Matheson, Jack Deslippe, Massimiliano Fatica, Prabhat, and Michael Houston. 2018. Exascale deep learning for climate analytics. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC 2018, Dallas, TX, USA, November 11-16, 2018.* IEEE / ACM, 51:1–51:12.

[14] Jakob Lüttgau, Shane Snyder, Philip H. Carns, Justin M. Wozniak, Julian M. Kunkel, and Thomas Ludwig. 2018. Toward Understanding I/O Behavior in HPC Workflows. In *3rd IEEE/ACM International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS@SC), Dallas, TX, USA, November 12.* 64–75. https://doi.org/10.1109/PDSW-DISCS.2018.00012

[15] Huong Luu, Babak Behzad, Ruth A. Aydt, and Marianne Winslett. 2013. A multi-level approach for understanding I/O activity in HPC applications. In *2013 IEEE International Conference on Cluster Computing, CLUSTER 2013, Indianapolis, IN, USA, September 23-27, 2013.* IEEE Computer Society, 1–5.

[16] D Miller, J Whitlocak, M Gartiner, M Ralphson, R Ratovsky, and U Sarid. [n. d.]. OpenAPI Specification v3. 1.0 (2021). *URL https://spec. openapis. org/oas/latest. html. OpenAPI Initiative, The Linux Foundation* ([n. d.]).

[17] Tirthak Patel, Suren Byna, Glenn K. Lockwood, and Devesh Tiwari. 2019. Revisiting I/O behavior in large-scale storage systems: the expected and the unexpected. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2019, Denver, Colorado, USA, November 17-19, 2019,* Michela Taufer, Pavan Balaji, and Antonio J. Peña (Eds.). ACM, 65:1–65:13.

[18] Arnab K. Paul, Jong Youl Choi, Ahmad Maroof Karimi, and Feiyi Wang. 2022. Machine Learning Assisted HPC Workload Trace Generation for Leadership Scale Storage Systems. In *31st International Symposium on High-Performance Parallel and Distributed Computing (HPDC), Minneapolis, MN, USA, 27 June 2022 - 1 July.* 199–212. https://doi.org/10.1145/3502181.3531457

[19] Ivy Bo Peng, Roberto Gioiosa, Gokcen Kestor, Jeffrey S. Vetter, Pietro Cicotti, Erwin Laure, and Stefano Markidis. 2018. Characterizing the performance benefit of hybrid memory system for HPC applications. *Parallel Comput.* 76 (2018), 57–69.

[20] Sameer Shende, Allen D. Malony, Wyatt Spear, and Karen Schuchardt. [n. d.]. Characterizing I/O Performance Using the TAU Performance System. In *Applications, Tools and Techniques on the Road to Exascale Computing, Proceedings of the conference ParCo 2011, 31 August - 3 September 2011, Ghent, Belgium (Advances in Parallel Computing, Vol. 22),* Koen De Bosschere, Erik H. D'Hollander, Gerhard R. Joubert, David A. Padua, Frans J. Peters, and Mark Sawyer (Eds.). IOS Press, 647–655.

[21] Shane Snyder, Philip Carns, Kevin Harms, Robert Ross, Glenn K Lockwood, and Nicholas J Wright. 2016. Modular hpc i/o characterization with darshan. In *2016 5th workshop on extreme-scale programming tools (ESPT).* IEEE, 9–17.

[22] Chen Wang, Jinghan Sun, Marc Snir, Kathryn M. Mohror, and Elsa Gonsiorowski. 2020. Recorder 2.0: Efficient Parallel I/O Tracing and Analysis. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2020, New Orleans, LA, USA, May 18-22, 2020.* IEEE, 1052–1059.

[23] Teng Wang, Shane Snyder, Glenn K. Lockwood, Philip H. Carns, Nicholas J. Wright, and Suren Byna. 2018. IOMiner: Large-Scale Analytics Framework for Gaining Knowledge from I/O Logs. In *IEEE International Conference on Cluster Computing (CLUSTER), Belfast, UK, September 10-13.* 466–476.

[24] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3 (2016).

[25] Cong Xu, Shane Snyder, Vishwanath Venkatesan, Philip Carns, Omkar Kulkarni, Suren Byna, Roberto Sisneros, and Kalyana Chadalavada. 2017. *Dxt: Darshan extended tracing.* Technical Report. Argonne National Lab.(ANL), Argonne, IL (United States).