ION: Navigating the HPC I/O Optimization Journey using Large Language Models

Arnav Sareen, UNC Charlotte

Chris Egersdoerfer and Dr. Dong Dai, College of Computing and Informatics

Introduction

- Scientific applications are increasingly data intensive
- The complicated nature of the HPC I/O stack and the number of configuration options makes it difficult for scientists to optimize I/O performance
- The adaptability of LLMs to domain-specific tasks make them well suited to analyzing application I/O traces for clues to improve performance
- **I/O Navigator** (**ION**) is an LLM-based framework to provide diagnosis summaries of common I/O issues in an approachable, accessible format

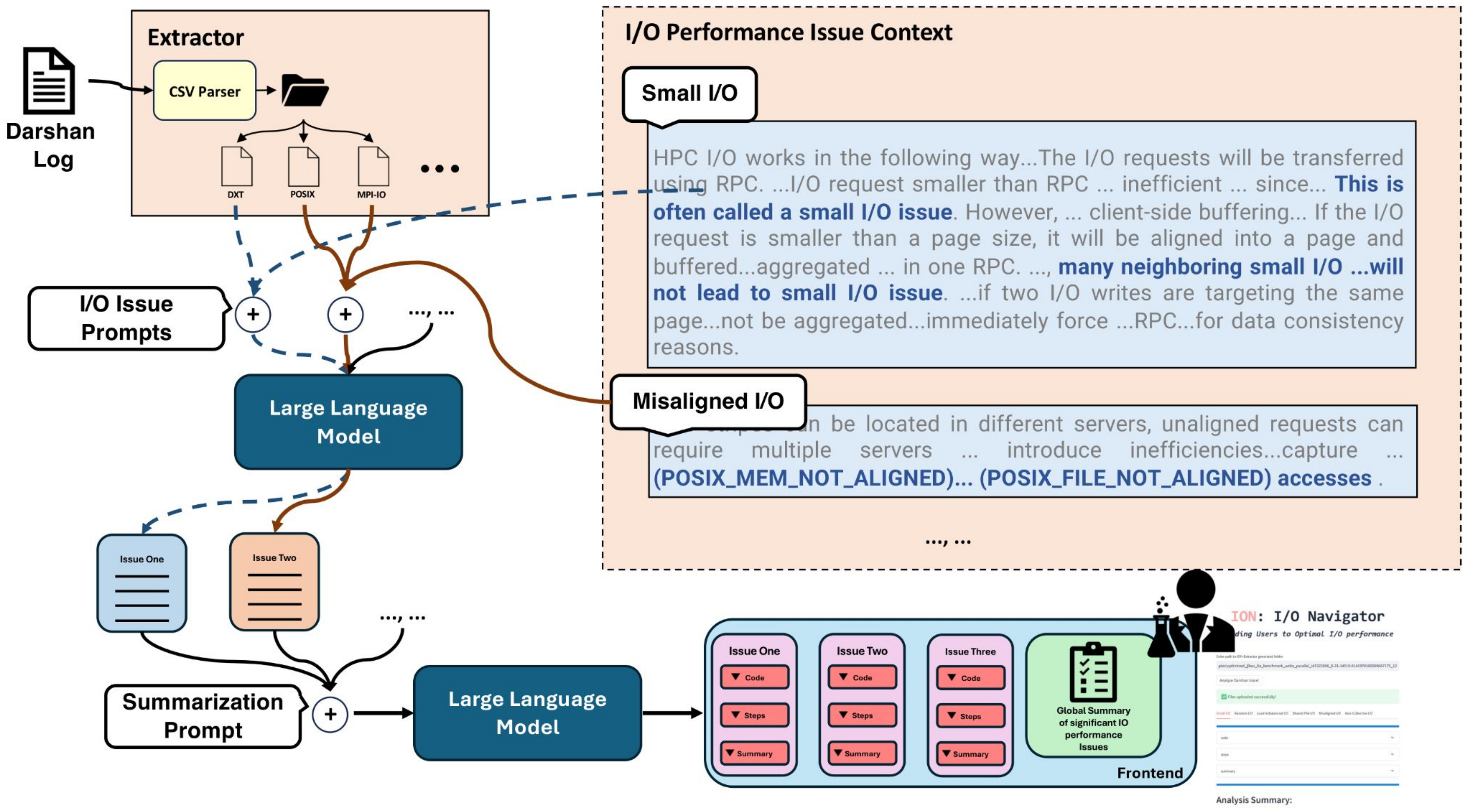
Design and Implementation

Extractor

- Unpacks and parses Darshan Log Files expected by the Analyzer
  - Each module of the log is divided into its own CSV file with the appropriate counters
- The CSV files are mapped to specific issues for analysis; not all issues need every CSV file
- A streamlined avenue for an efficient, timely analysis by the LLM

Analyzer

- Carefully engineered prompts that provide vital context combined with appropriate CSV files are sent to the LLM (*gpt-4-1106-preview*)
- Utilizes a *divide-and-conquer* approach that focuses upon one issue at a time and then combines the diagnoses into a comprehensive summary
- There is a message window to allow users to interact with ION just as they would converse with a human expert



Small I/O: HPC I/O works in the following way...The I/O requests will be transferred using RPC. ...I/O request smaller than RPC ... inefficient ... since... **This is often called a small I/O issue**. However, ... client-side buffering... If the I/O request is smaller than a page size, it will be aligned into a page and buffered...aggregated ... in one RPC. ..., **many neighboring small I/O ...will not lead to small I/O issue**. ...if two I/O writes are targeting the same page...not be aggregated...immediately force ...RPC...for data consistency reasons.

Misaligned I/O: …n be located in different servers, unaligned requests can require multiple servers ... introduce inefficiencies...capture ... **(POSIX_MEM_NOT_ALIGNED)... (POSIX_FILE_NOT_ALIGNED) accesses** .

Figure 1: The overall workflow of ION and its key Components

Background

Key Technologies

- **Large Language Models (LLMs)** - AI systems trained on extensive datasets that can interact with human language
- **Darshan** - Tracks important metrics from I/O traces
- **Drishti** - Tool that identifies performance issues in Darshan logs via predefined thresholds
- **In-context Learning -** Providing further context to LLMs in the prompt
- **Chain of Thought (CoT**) - Enabling LLMs to think step-by-step

Previous Work

- **LLMs as domain experts** - LLMs can excel at domain-specific tasks via in-context learning
- **LLMs for Log Analysis** - Largely limited thus far to system and cloud logs, not HPC logs

Evaluation

IO500 Benchmark

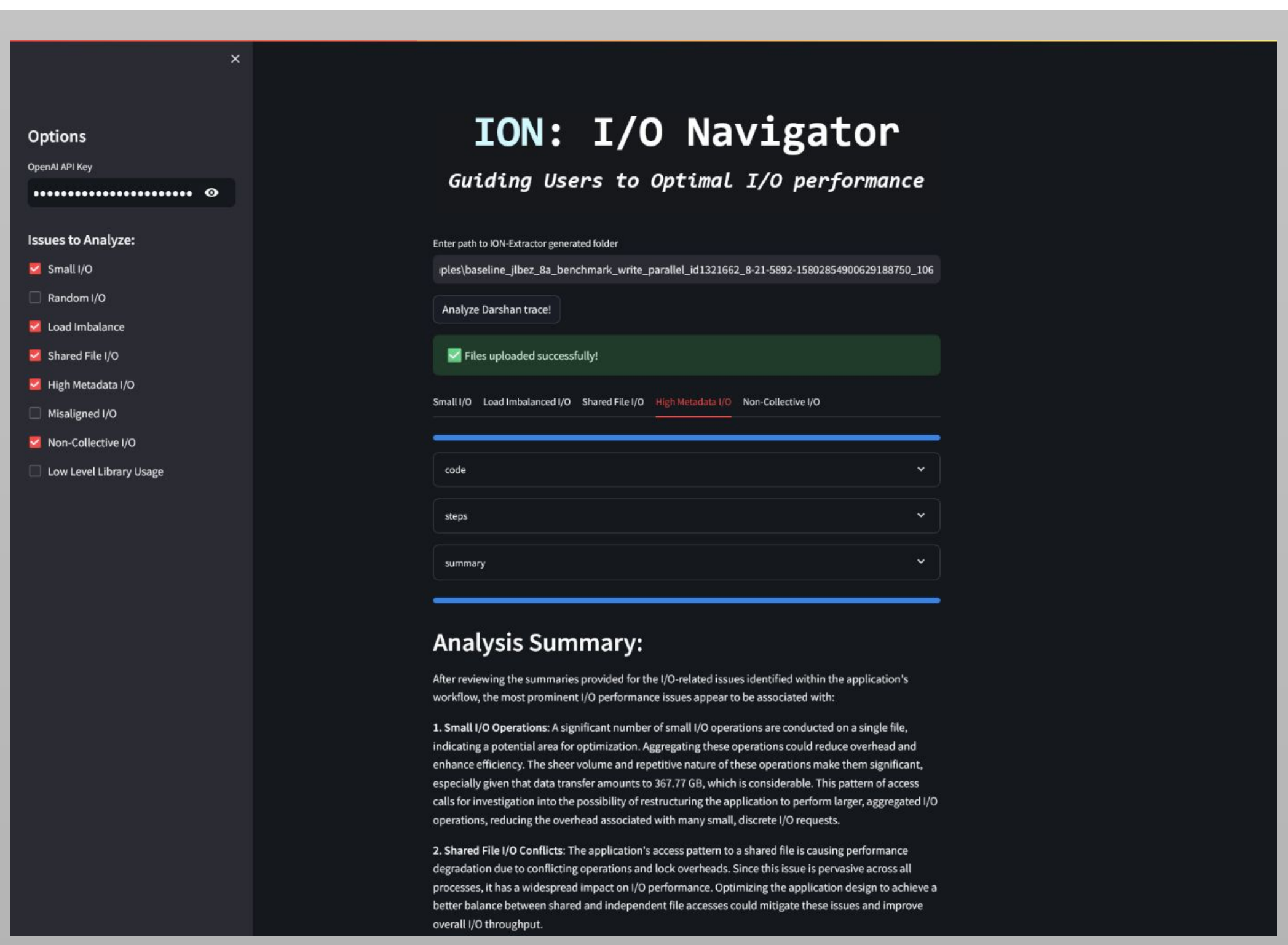
- Injected specific, known issues into the **IO500 benchmark** to test ION versus a set of ground truths
- ION can correctly identify the root diagnosis of these issues and provide a user with the justification for it

Real Applications

- Compared to Drishti, ION is able to provide specific solutions tailored to the inputted application (*Figure 2*)
- ION is also able to conglomerate I/O issues together to paint a holistic picture of I/O performance

	ION Output	Drishti Output
OpenPMD (Baseline)	Application demonstrates a potentially inefficient I/O pattern…**98.78% of operations being small I/O that might lead to underutilization of RPCs**… **as most small I/O are consecutive, aggregation optimization might be in place**…Significant file misalignment detected…this may contribute to performance degradation due to increased contention… .(..8a_parallel_3Db_0000001.h5) is being accessed by multiple ranks (384 in total)…	issues a high number (275840) of small read requests (i.e., < 1MB).. issues a high number (427386) of small write requests (i.e., < 1MB)…(64.38%) small write requests are to "8a_parallel_3Db_0000001.h5" …Application issues a high number (100.00%) of misaligned file requests…
OpenPMD (Optimized)	The trace file indicates that 3.31% of total I/O operations consist of small-sized operations, **with 88.15% of these exhibiting random access patterns… However, the random access operation count per rank and the total volume of data transferred through these patterns are low**…not affecting the entire application's I/O performance.	Application is issuing a high number (565) of random read operations (35.25%)…

Figure 2: Comparison of Drishti and ION for OpenPMD application



ION: I/O Navigator

Guiding Users to Optimal I/O performance

Enter path to ION-Extractor generated folder

iples\baseline_jlbez_8a_benchmark_write_parallel_id1321662_8-21-5892-15802854900629188750_106

✅ Files uploaded successfully!

Analysis Summary:

After reviewing the summaries provided for the I/O-related issues identified within the application's workflow, the most prominent I/O performance issues appear to be associated with:

1. Small I/O Operations: A significant number of small I/O operations are conducted on a single file, indicating a potential area for optimization. Aggregating these operations could reduce overhead and enhance efficiency. The sheer volume and repetitive nature of these operations make them significant, especially given that data transfer amounts to 367.77 GB, which is considerable. This pattern of access calls for investigation into the possibility of restructuring the application to perform larger, aggregated I/O operations, reducing the overhead associated with many small, discrete I/O requests.

2. Shared File I/O Conflicts: The application's access pattern to a shared file is causing performance degradation due to conflicting operations and lock overheads. Since this issue is pervasive across all processes, it has a widespread impact on I/O performance. Optimizing the application design to achieve a better balance between shared and independent file accesses could mitigate these issues and improve overall I/O throughput.

Figure 3: ION Web Interface

Conclusions

- In this poster, we explain ION, its overall workflow and ability to match state-of-the-art solutions
- Future improvements to ION include building a holistic knowledge base of HPC I/O to allow the LLM to make more accurate decisions and provide better suggestions

References

Philip Carns, Kevin Harms, William Allcock, Charles Bacon, Samuel Lang, Robert Latham, and Robert Ross. 2011. Understanding and Improving Computational Science Storage Access through Continuous Characterization. *ACM Trans. Storage* 7, 3, Article 8 (oct 2011), 26 pages. https://doi.org/10.1145/2027066.2027068

Jean Luca Bez, Hammad Ather, and Suren Byna. 2022. Drishti: Guiding End-Users in the I/O Optimization Journey. *In 2022 IEEE/ACM International Parallel Data Systems Workshop (PDSW)*. 1–6. https://doi.org/10.1109/PDSW56643.2022.00006