DIRLAB

Hot Storage

ION

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

Chris Egersdoerfer, Dong Dai | *University of Delaware*
Arnav Sareen | *University of North Carolina at Charlotte*
Jean Luca Bez | *Lawrence Berkeley National Lab*
Suren Byna | *The Ohio State University*

CHARLOTTE
COLLEGE OF COMPUTING AND INFORMATICS

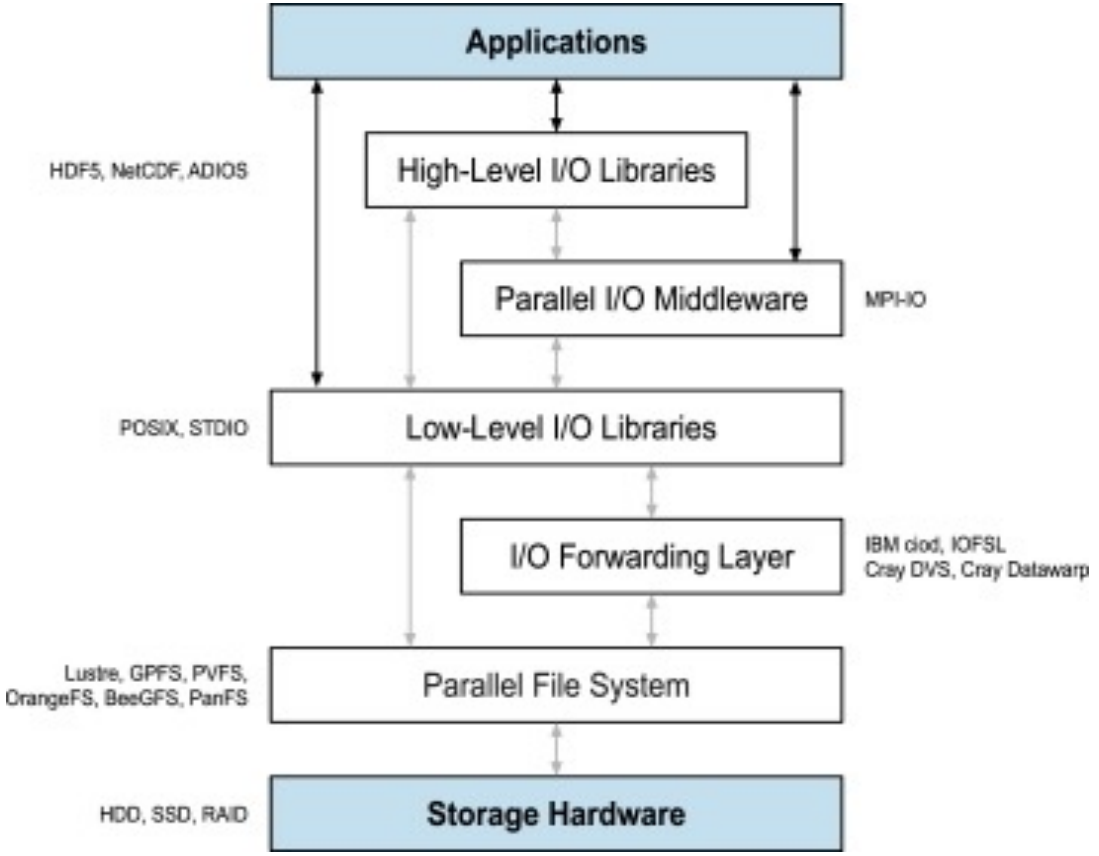
BERKELEY LAB

THE OHIO STATE UNIVERSITY

High Level Introduction: HPC I/O stack

- The HPC I/O stack is complex
- Many levels
  - High-Level I/O
  - I/O middleware
  - Low-Level I/O
  - I/O forwarding
  - Parallel File Systems
- Various options at each level
- Many interacting parameters

- **It is challenging for users/developers leverage all layers effectively**



DIRLAB

Hot Storage

Correcting I/O behavior

- Many common pitfalls
  - Improper use of parallel I/O middleware
  - I/O request misalignment
  - Inefficient read/write patterns
  - Shared files between ranks
  - Load imbalance among ranks

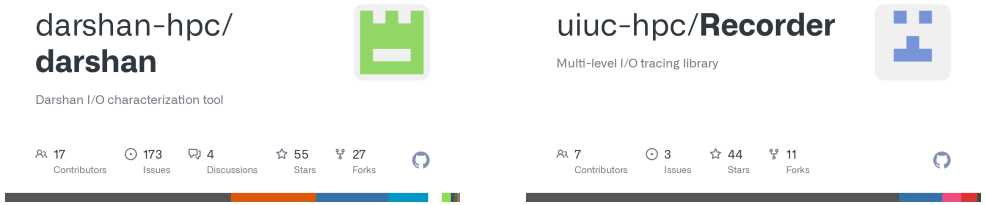
Performance Impact

- E2E (domain decomposition kernel)
  - Fixing significant load imbalance led to 10x* speedup

* 10.1109/PDSW56643.2022.00006

Current Solutions

- How do we diagnose and resolve I/O inefficiencies?
  - Profiling tools create detailed trace logs
    - Darshan
    - Recorder



Profilers

- Analysis tools extract/visualize key metrics
  - PyDarshan
  - DXT-Explorer

PyDarshan

DXT EXPLORER

Analysis tools

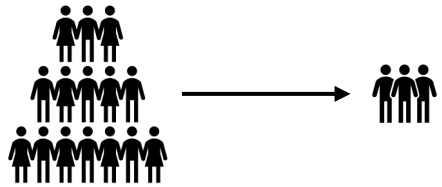
- Diagnose tools indicate potential performance issues
  - Drishti

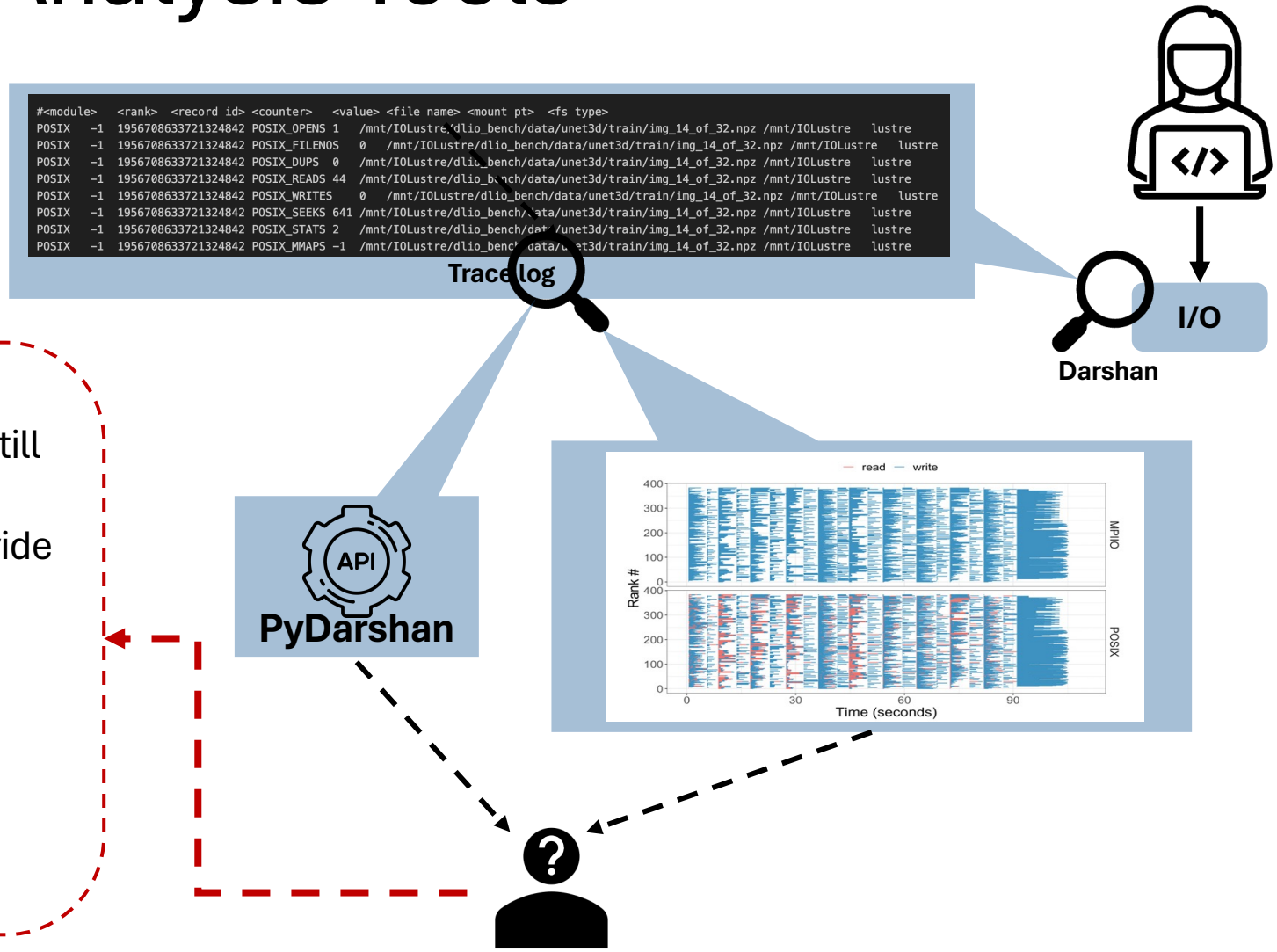
Drishti

Diagnosis tools

Analysis Tools

- Turning analysis into actionable insights still requires domain expertise
  - Few I/O experts are available to provide expertise

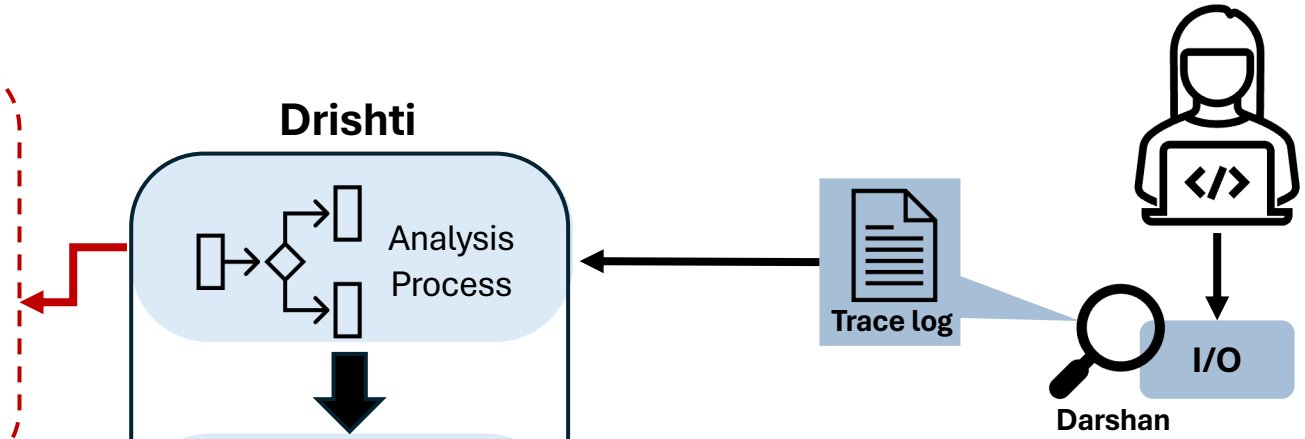




Diagnosis Tools

- Trigger-based
  - Threshold settings require domain expertise
    - % of time doing metadata ops,
    - % of operations which should be collective

- Generic outputs
  - Static code samples
  - Minimal explanation
- Ignores overlapping factors
  - Requires expertise to properly interpret diagnoses




```
METADATA

  ▶ Application is read operation intensive (6.34% writes vs. 93.66% reads)
  ▶ Application might have redundant read traffic (more data was read than the highest read offset)
  ▶ Application might have redundant write traffic (more data was written than the highest write offset)

OPERATIONS

  ▶ Application issues a high number (285) of small read requests (i.e., < 1MB) which represents 37.11% of all
read/write requests
    ↳ 284 (36.98%) small read requests are to "benchmark.h5"
  ▶ Application mostly uses consecutive (2.73%) and sequential (90.62%) read requests
  ▶ Application mostly uses consecutive (19.23%) and sequential (76.92%) write requests
  ▶ Application uses MPI-IO and read data using 640 (83.55%) collective operations
  ▶ Application uses MPI-IO and write data using 768 (100.00%) collective operations
  ▶ Application could benefit from non-blocking (asynchronous) reads
  ▶ Application could benefit from non-blocking (asynchronous) writes
  ▶ Application is using inter-node aggregators (which require network communication)
```

LLM Capabilities

- LLM capabilities have advanced significantly
  - Long context
  - Robust instruction following

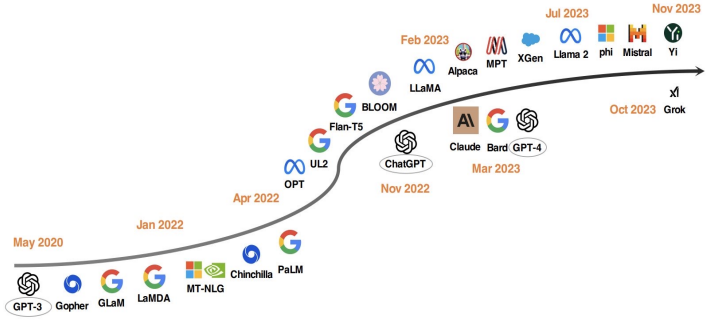
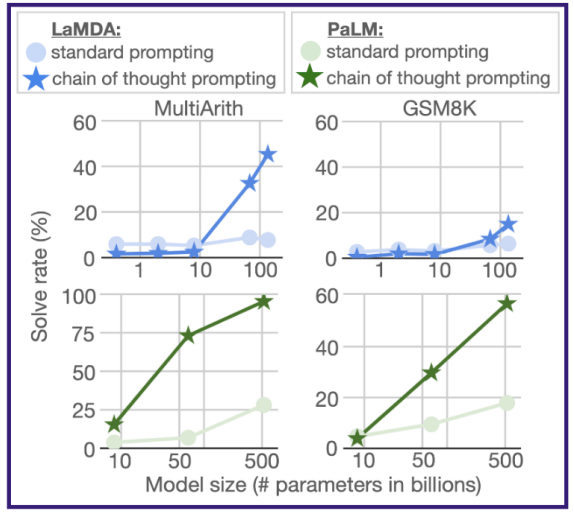


Figure 3: LLM development timeline. The models below the arrow are closed-source while those above the arrow are open-source.

- Post-training techniques have emerged
  - Chain of Thought
  - One/Few-shot prompting
  - Retrieval-Augmented Generation
  - LLMs as agents



Can LLMs interpret profiling logs?

Problems

- Frontier LLMs are general models
  - Lack robust/detailed knowledge of HPC
- LLMs naturally hallucinate

Solutions

- Provide robust/comprehensive **descriptions** of various I/O issues for in-context learning
- Utilize **Chain of Thought** (CoT) prompting
  - Boosts consistency and accuracy

Small I/O Context

If an I/O request is smaller than the RPC size, it will be aggregated with others before being sent to the OSTs if they are sequential

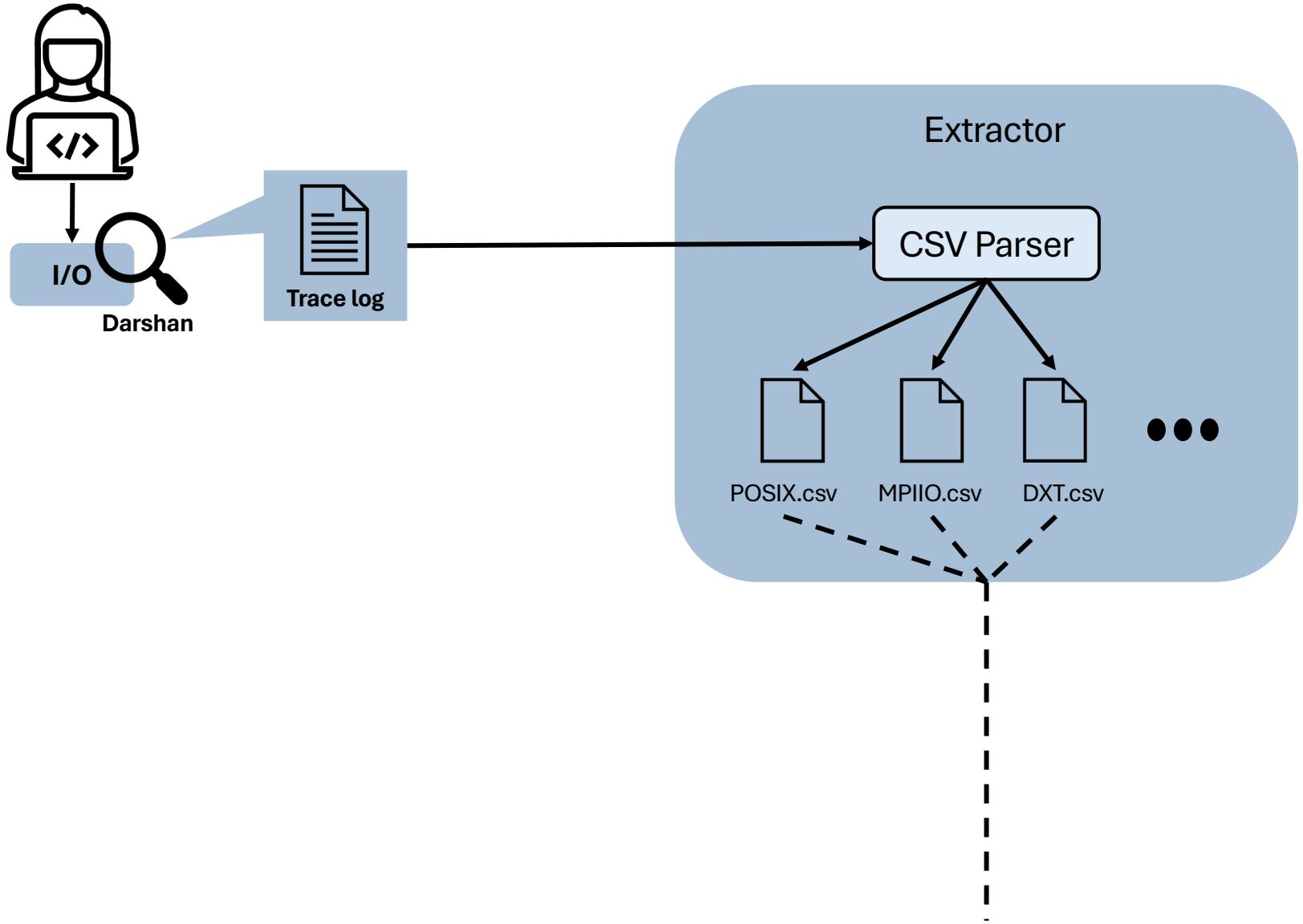
...

can mostly be ignored if the application only accesses a file once or twice via small I/O requests

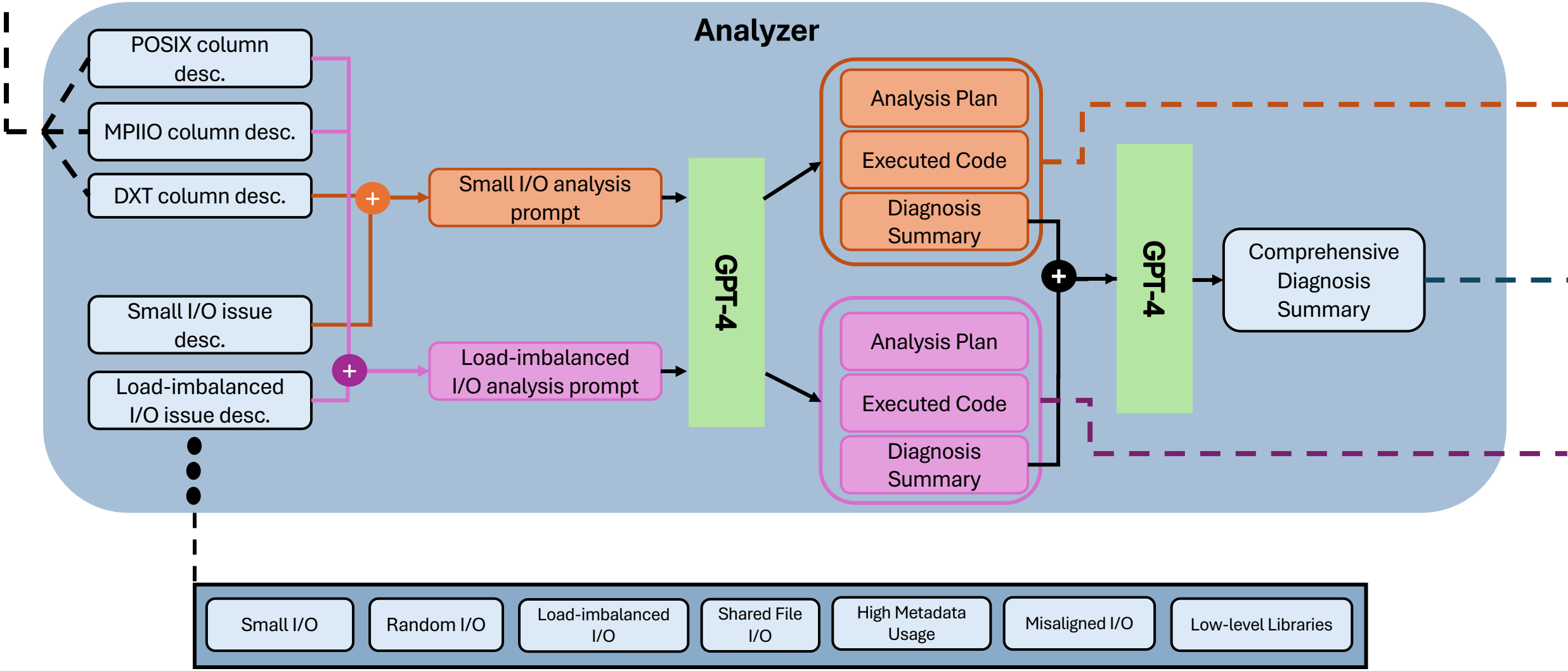
...

To diagnose the issue, first analyze the size of the I/O requests, then check to see if any of the requests identified as small are accessing the same files multiple times, and finally check if any of the small requests might be aggregated based on their access patterns.

ION Design


I/O
Darshan
Trace log
Extractor
CSV Parser
POSIX.csv
MPIIO.csv
DXT.csv

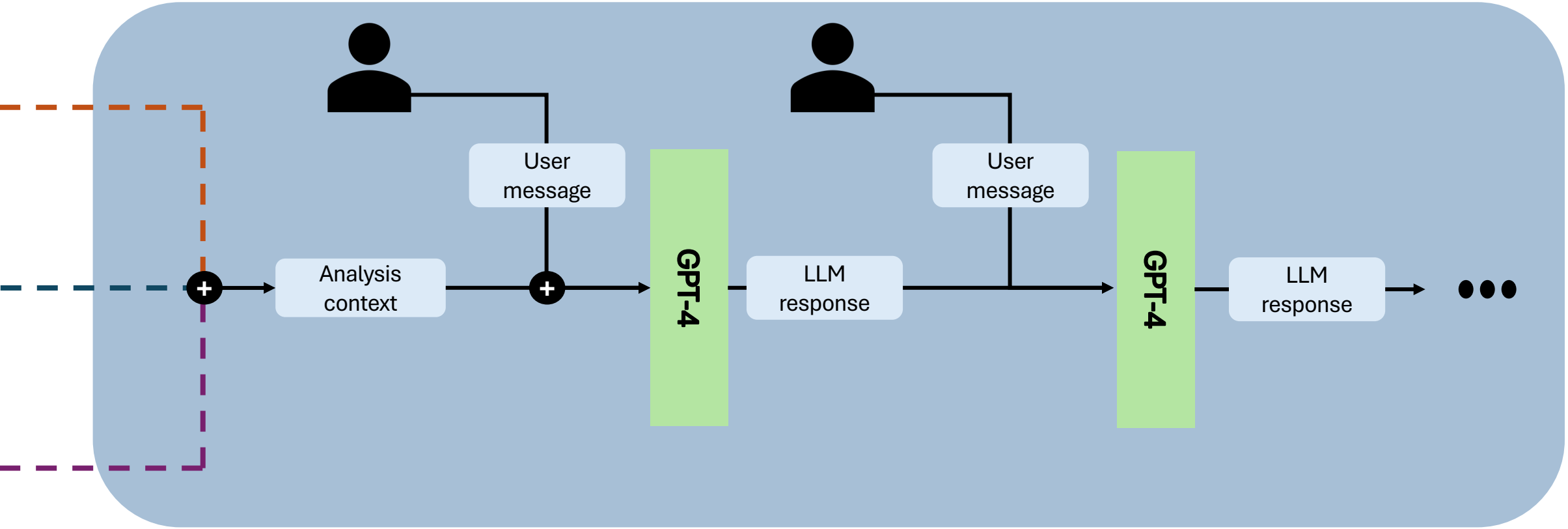
Design Cont...


Analyzer
POSIX column desc.
MPIIO column desc.
DXT column desc.
Small I/O issue desc.
Load-imbalanced I/O issue desc.
Small I/O analysis prompt
Load-imbalanced I/O analysis prompt
GPT-4
Analysis Plan
Executed Code
Diagnosis Summary
Analysis Plan
Executed Code
Diagnosis Summary
GPT-4
Comprehensive Diagnosis Summary
Small I/O
Random I/O
Load-imbalanced I/O
Shared File I/O
High Metadata Usage
Misaligned I/O
Low-level Libraries

DIRLAB

Hot Storage

Design Cont...



DIRLAB

Hot Storage

Evaluation

IO500

	Ground Truth	ION Outputs
IOR-Easy-2KB-Shared-File	• Small read/write requests • Misaligned to stripe • Sequential and Consecutive. • 4 ranks read/write to a single shared file using Posix api	...I/O operations are **Small and Target the same file,... largely sequential**, which **could allow for aggregation and mitigate some inefficiencies**...Significant file **misalignment detected affecting 99.80% of I/O operations**...application is only using **POSIX I/O calls and not employing MPI-IO, despite the presence of multiple ranks performing I/O**...
IOR-Easy-1MB-File-per-process	• Small read/write requests • Sequential and Consecutive • 4 ranks read/write to there own files using posix api	...**Repetitive small I/O**...significant majority (8184) of these small I/O requests are **potentially aggregatable... mitigate the inefficiency**,...each of the four unique files is accessed exclusively by a single rank...**does not exhibit any misaligned I/O behavior**...consecutive and sequential, suggesting that **the application could benefit from the use of MPI-IO's collective and non-blocking operations...does not use the MPI-IO module, ...**
IOR-Random-4K-Shared-File	• Small random read/write which cannot be aggregated. • Misaligned requests • 4 ranks read/write to shared file using Posix api	... **small and non-sequential..., there could be a performance concern related to random access patterns**...The POSIX_FILE_NOT_ALIGNED counter indicates **286,442 instances of file misalignment**...approximately 99.61% of the operations..., **but it does not use MPI-IO for I/O operations**

DIRLAB

Hot Storage

Evaluation Cont...

E2E

(Domain decomposition kernel)

	Unoptimized	Optimized
Ground Truth	• Mishandling of NetCDF caused significant load imbalance on rank 0 • File misalignment	• Primary load imbalance is resolved • File misalignment persists
ION	**Misaligned memory accesses…(approximately 99.8%) of the write operations are misaligned**…Rank 0 also has much larger summed I/O sizes …, which indicates that rank 0 is doing much more work…	**…a pervasive issue with file access alignment, with 99.8% of file I/O operations being misaligned**. … A subset of **64 out of the 1024 ranks exhibit a significantly higher number of I/O operations per second**…their throughput stats far exceeding one standard deviation above the mean…these ranks contribute to approximately 98.23% of the total write operations.
Drishti	**Application issues a high number (99.81%) of misaligned file requests**…Load imbalance of 99.90% detected while accessing "3d_32_32_16_32_32_32.nc4”…	Application issues a high number (99.80%) of misaligned file requests…

Conclusion

ION

- LLM-based HPC I/O performance diagnosis tool
  1. Leverages in-context learning
  2. Conducts automated analysis of common HPC I/O performance issues
  3. Delivers a comprehensive and interpretable analysis summary based on reasoning
  4. Allows for natural user interaction

Future Work

- Escape the need for curated I/O issue contexts
- Remove constraints of pre-defined issues/analysis pathways
- Explore various domain knowledge alignment techniques
  - Fine-tuning

DIRLAB

Hot Storage

Q/A

CHARLOTTE
COLLEGE OF COMPUTING AND INFORMATICS

BERKELEY LAB

THE OHIO STATE UNIVERSITY