DIRLAB

SC24 Atlanta, GA | hpc creates.

Leveraging AI: Large Language Models in HPC I/O Optimization

Dong Dai

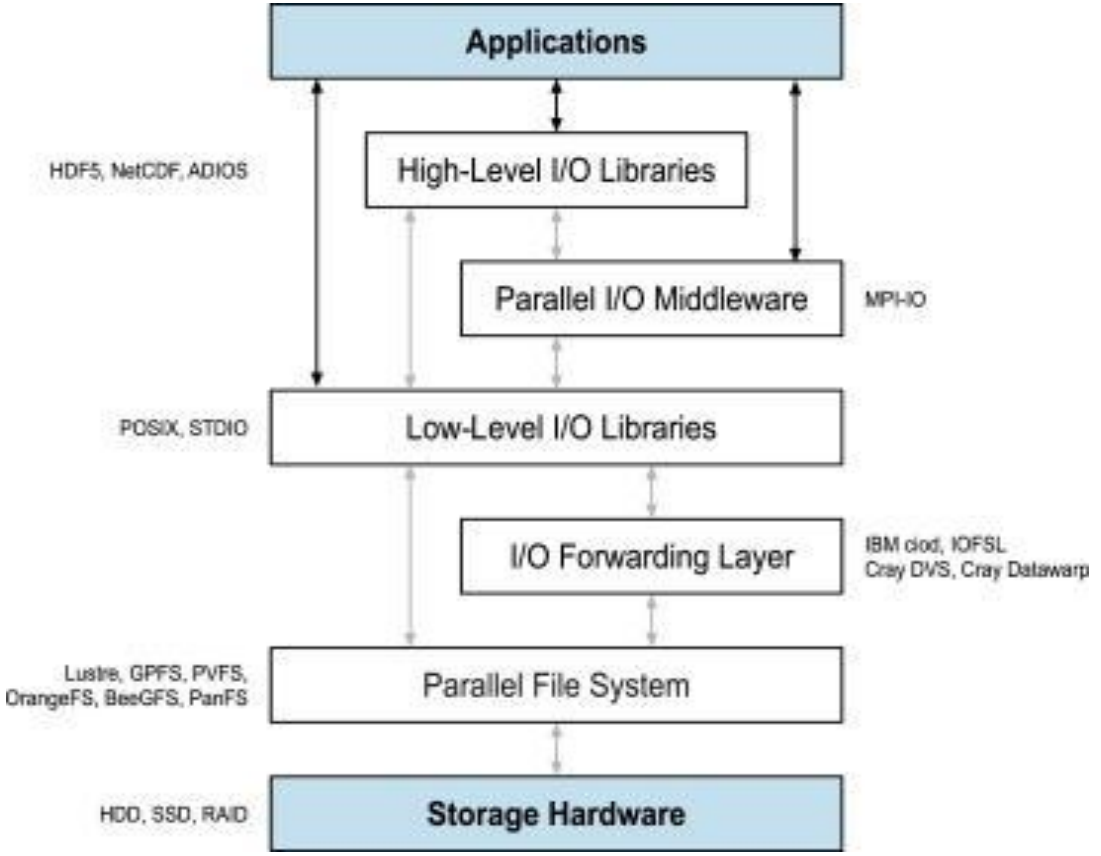
Associate Professor, University of Delaware

UNIVERSITY OF DELAWARE

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

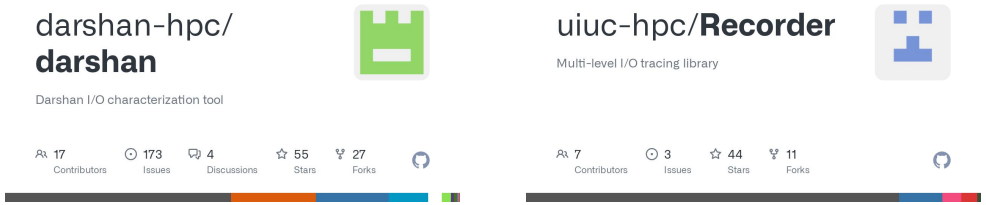
- **It is challenging for users/developers leverage all layers effectively**



Bez, Jean Luca, Suren Byna, and Shadi Ibrahim. "I/o access patterns in hpc applications: A 360-degree survey." ACM Computing Surveys 56, no. 2 (2023): 1-41.

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

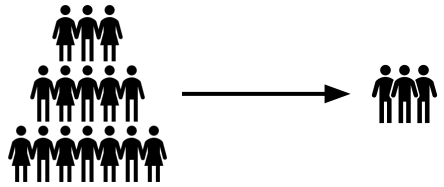
  - Diagnose tools indicate potential performance issues
    - Drishti

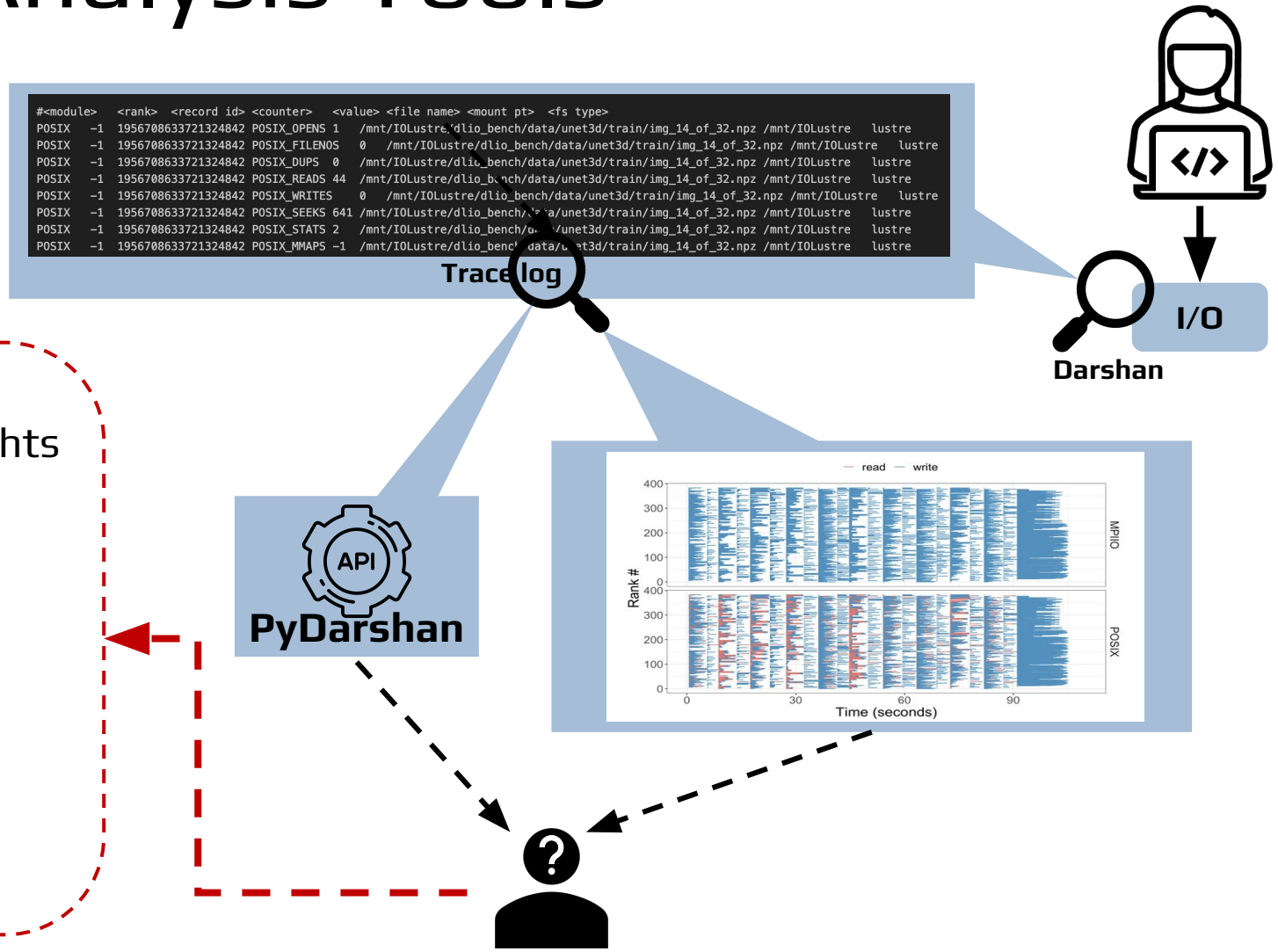
Drishti

Diagnosis tools

Analysis Tools

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

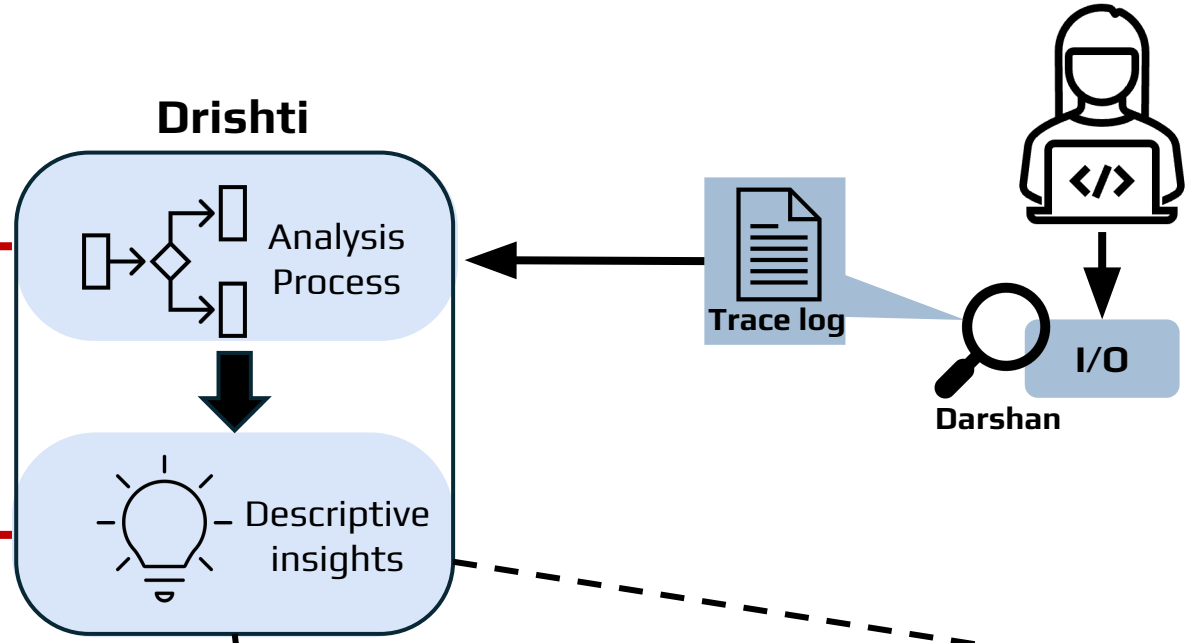




Diagnosis Tools

- 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
- May 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)
```

DIRLAB

SC24 Atlanta, GA | hpc creates.

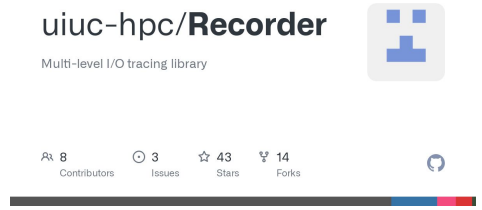
I/O Trace analysis

- Many profiling tools exist
- Domain scientists may not have expertise to interpret their output
- Not enough experts to help individual users
- Users need to wait long for questions to be answered

Can LLMs help democratize I/O performance Diagnosis?

They would need to be:

- Approachable
  - Any type of user should understand
- Interactive
  - Users can ask specific questions
- Accurate
  - Diagnosis content should be correct




```
#<module>   <rank>  <record id> <counter>   <value> <file name> <mount pt>  <fs type>
MPI-IO  -1  1448033504430027070 MPIIO_INDEP_OPENS   0   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_COLL_OPENS    8   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_INDEP_READS   12  /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_INDEP_WRITES  3   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_COLL_READS    0   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_COLL_WRITES   24  /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_SPLIT_READS   0   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_SPLIT_WRITES  0   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_NB_READS  0   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_NB_WRITES 0   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_SYNCS 0   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_HINTS 48  /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_VIEWS 48  /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
MPI-IO  -1  1448033504430027070 MPIIO_MODE  8   /pscratch/h5bench/build-offsets/plt00003.h5 /pscratch   lustre
```

Challenges

- Log content complexity
  - Requires domain knowledge to interpret
- Log size
  - LLM context windows are limited
- Performance issue diagnosis requires domain expertise t identify and diagnose

LLM Capabilities

- LLM capabilities have advanced significantly
  - Handle complex context
  - Instruction following

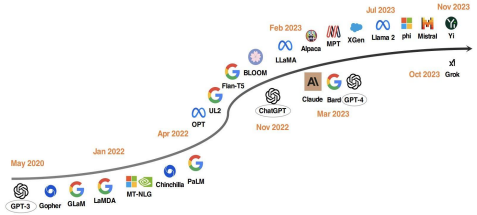
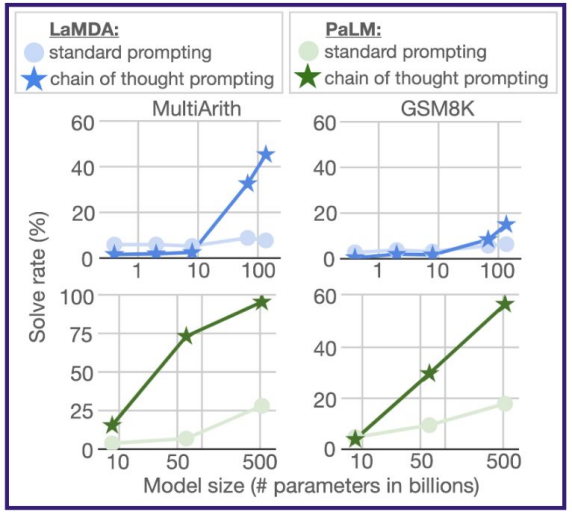


Figure 3: LLM development timeline. The models below the arrow are closed-source while those above the arrow are open-source.

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



LLM Capabilities

- LLM capabilities have advanced significantly
  - Handle complex context
  - Instruction following

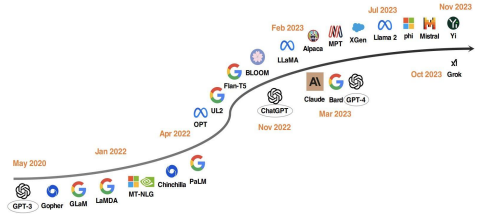
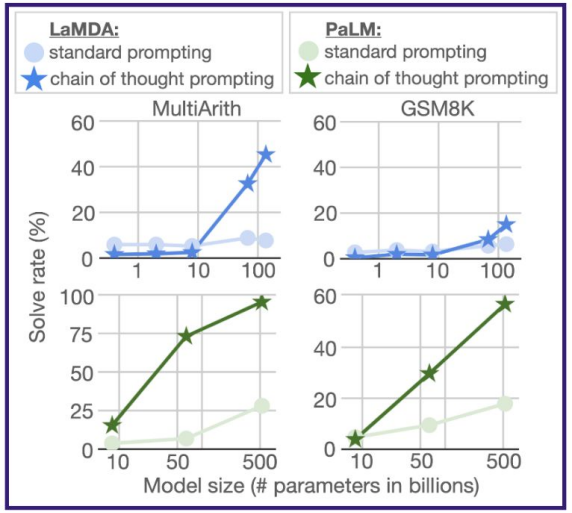
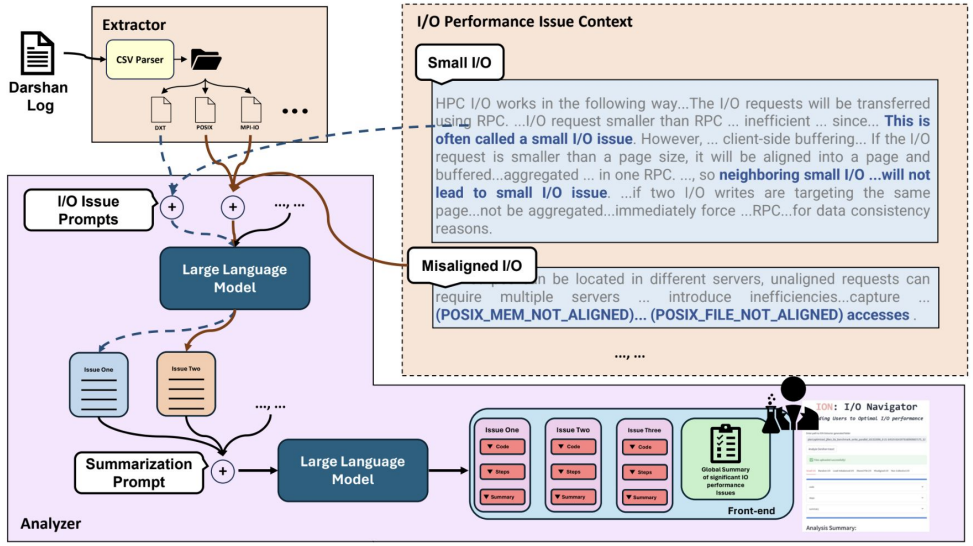


Figure 3: LLM development timeline. The models below the arrow are closed-source while those above the arrow are open-source.

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





	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%)...
E2E (Baseline)	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...	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"...
E2E (Optimized)	...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...it is worth investigating further to determine if this behavior is intentional (e.g., based on the application algorithm) or if it can be optimized for better load distribution.	Application issues a high number (99.80%) of misaligned file requests...

Egersdoerfer, Chris, Arnav Sareen, Jean Luca Bez, Suren Byna, and Dong Dai. "ION: Navigating the HPC I/O Optimization Journey using Large Language Models." In *Proceedings of the 16th ACM Workshop on Hot Topics in Storage and File Systems*, pp. 86-92. 2024

Filling the Gaps: Accuracy and Scalability

Log size

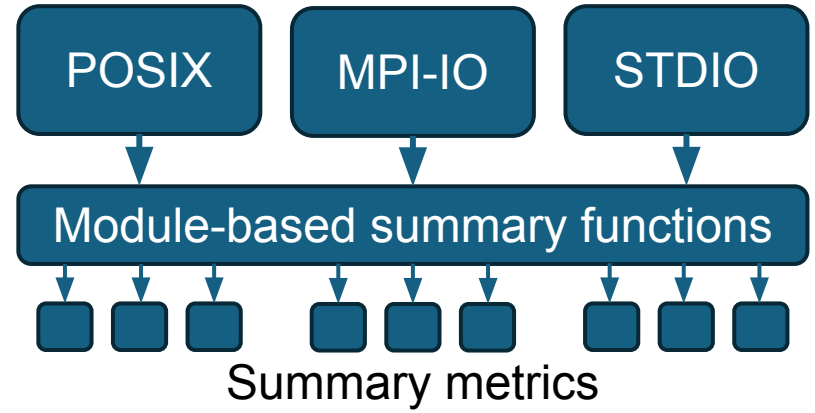
- Unpacked darshan logs for scientific applications can be many MB

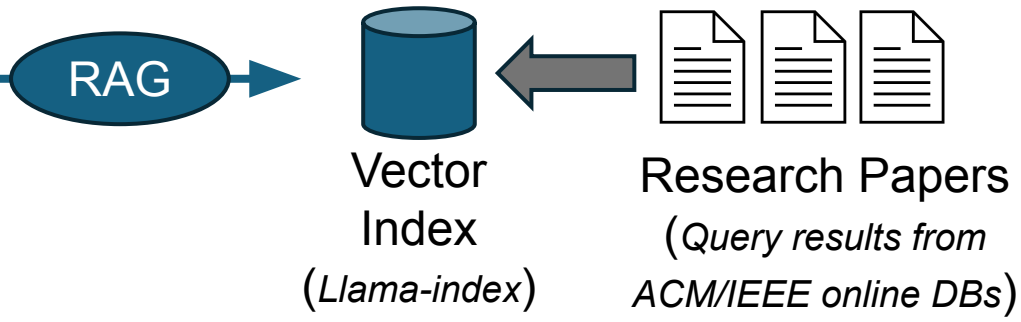
Log content complexity

- Darshan counters are per-file
- Analysis often requires aggregation of counters
  - LLMs cannot aggregate counters during generation

Performance issue diagnosis requires domain expertise

- Interpreting counter values to diagnose performance issues is non-trivial



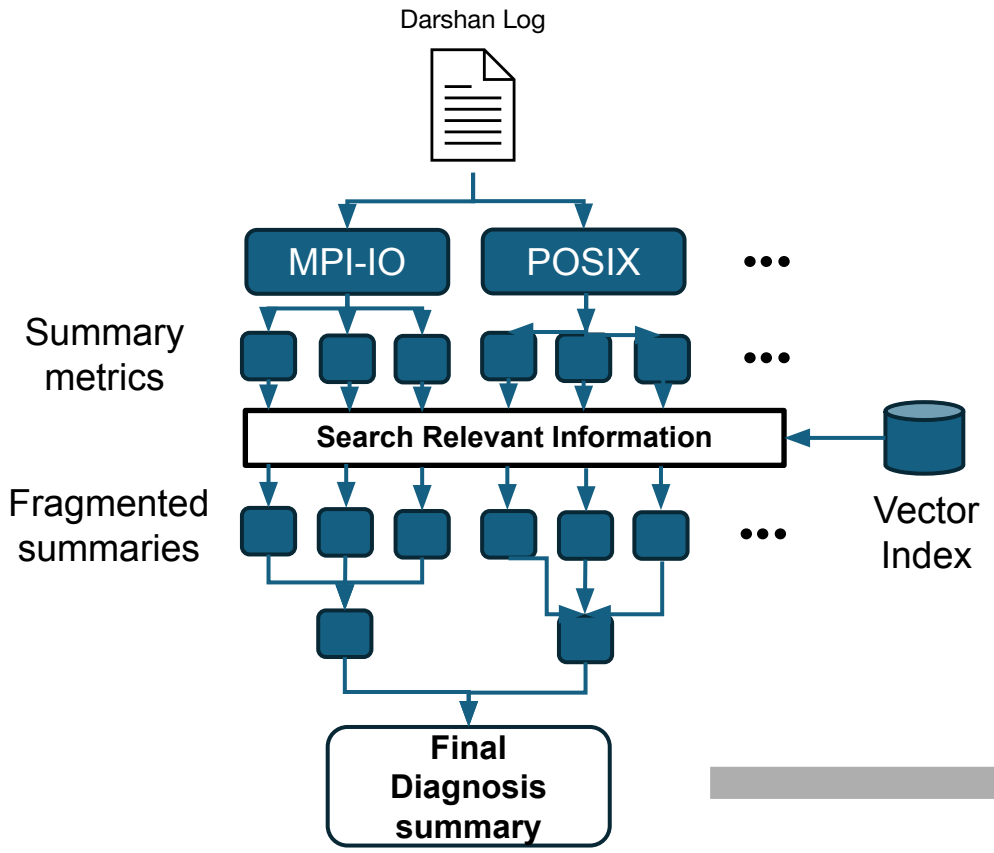


DIRLAB

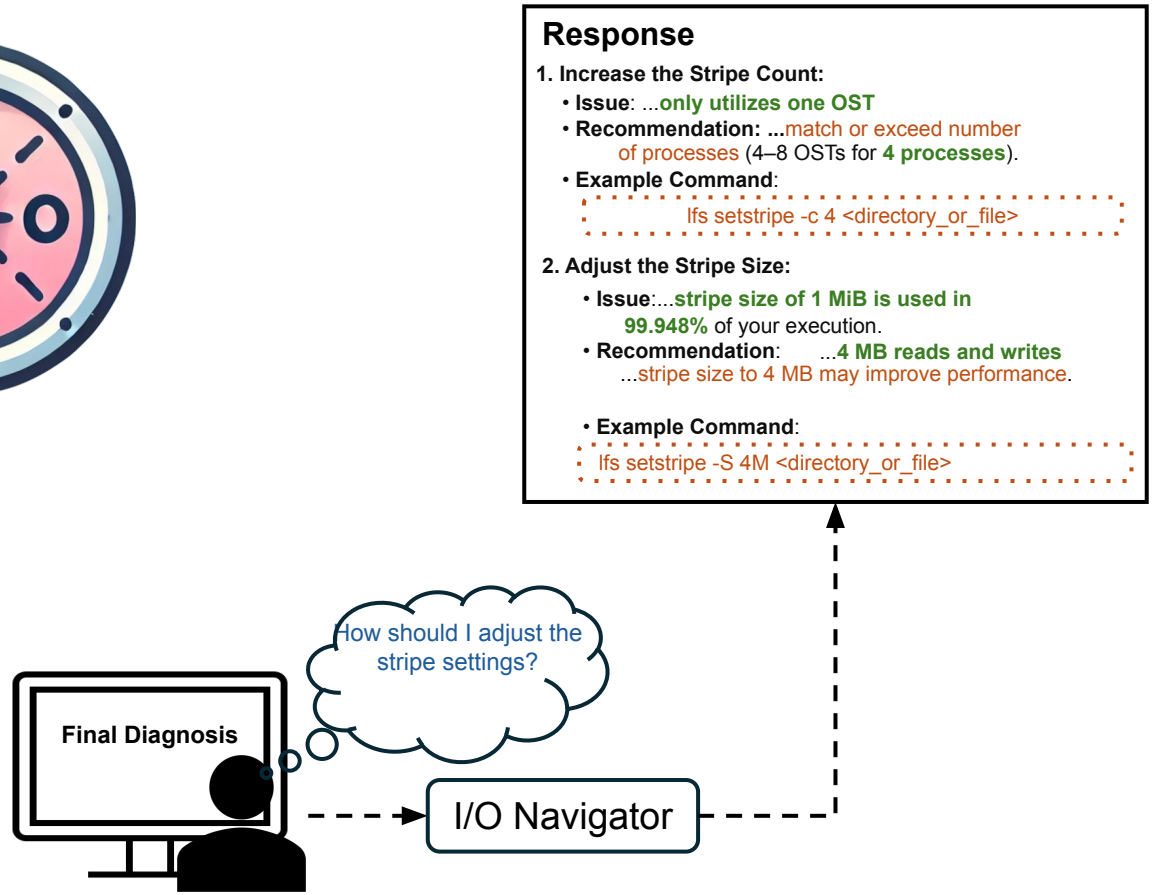
SC24 Atlanta, GA | hpc creates.

I/O Navigator

Workflow



User Interaction



DIRLAB

SC24 Atlanta, GA | hpc creates.

Some Results

- *We built a new benchmark suite to evaluate the system*

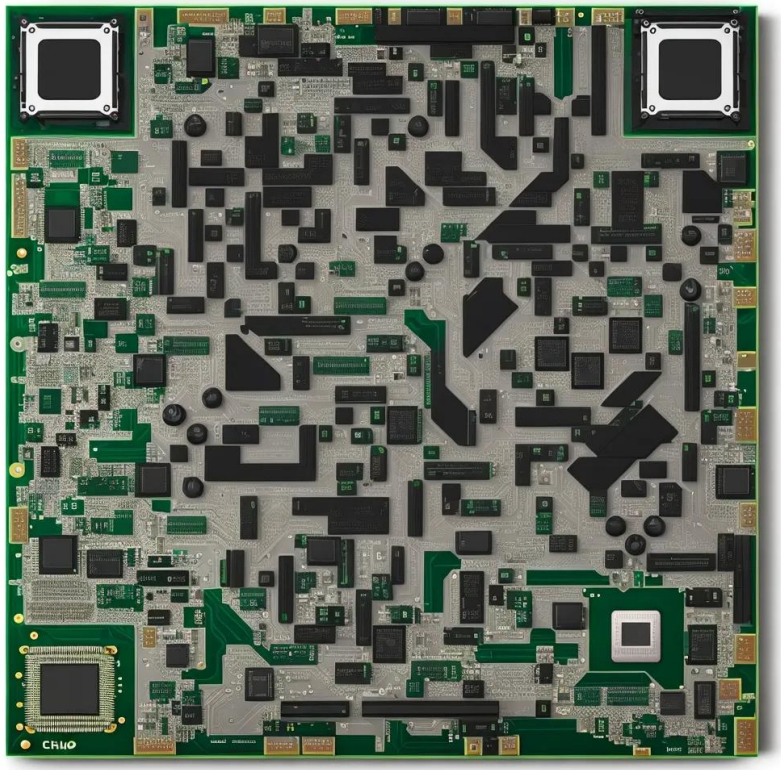
Labeled Issue	SB	IO500	RA	Total
High Metadata Load	1	2	2	5
Misaligned Read requests	2	10	4	16
Misaligned Write requests	2	10	6	18
Random Access Patterns on Write	0	5	2	7
Random Access Patterns on Read	0	5	2	7
Shared File Access	1	14	4	19
Small Read I/O Requests	2	10	5	17
Small Write I/O Requests	2	10	6	18
Repetitive Data Access on Read	1	0	0	1
Server Load Imbalance	7	15	2	24
Rank Load Imbalance	1	0	1	2
Multi-Process W/O MPI	0	13	0	13
No Collective I/O on Read	6	8	4	18
No Collective I/O on Write	5	8	2	15
Low-Level Library on Read	1	0	0	1
Low-Level Library on Write	1	0	0	1

Metric	Diagnosis Tool	Simple-Bench	IO500	Real-Applications	Overall
Accuracy	Drishti	0.398	0.480	0.472	0.459
	ION	0.343	0.381	0.417	0.380
	IOAgent-gpt-4o	**0.630**	**0.655**	**0.620**	**0.641**
	IOAgent-llama-3.1-70B	0.620	0.488	0.463	0.513
Utility	Drishti	0.426	0.417	0.491	0.436
	ION	0.352	0.401	0.380	0.385
	IOAgent-gpt-4o	0.565	**0.615**	**0.639**	**0.609**
	IOAgent-llama-3.1-70B	**0.694**	0.587	0.565	0.607
Interpretability	Drishti	0.417	0.452	0.463	0.447
	ION	0.343	0.417	0.352	0.385
	IOAgent-gpt-4o	0.546	**0.659**	**0.713**	**0.645**
	IOAgent-llama-3.1-70B	**0.694**	0.484	0.472	0.530
Average	Drishti	0.414	0.450	0.475	0.447
	ION	0.346	0.399	0.383	0.383
	IOAgent-gpt-4o	0.580	**0.643**	**0.657**	**0.632**
	IOAgent-llama-3.1-70B	**0.670**	0.520	0.500	0.550

DIRLAB

SC24 Atlanta, GA | hpc creates.

Scan to try out our demo!



UNIVERSITY OF DELAWARE

BERKELEY LAB
Lawrence Berkeley National Laboratory

THE OHIO STATE UNIVERSITY

NC STATE UNIVERSITY